Memoria Trabajo fin de carrera

Proyecto: Aplicación Web para la Gestión de la Formación - GESFOR

Alumno: David Colmenar de Viedma

Director de proyecto: Vicenç Font Sagrista

Índice General

1 ESTUDIO DE VIABILIDAD DEL SISTEMA

1.1 Necesidad de la formación interna en la empresa

La rutina diaria, el cumplimiento de objetivos económicos y la teórica "falta de tiempo" nos lleva a dejar en último plano algo tan importante como es el reciclaje del conocimiento, la formación continua de los trabajadores/as.

La formación no es una pérdida de tiempo o un gasto para la empresa sino todo lo contrario, es una inversión a largo plazo.

Por esto, no sólo la empresa tiene que entender la formación como una inversión, también el propio trabajador debe valorarlo así, teniendo en cuenta que por un lado va a aportar un mayor valor a su empresa y por otro va a aumentar notablemente su enriquecimiento intelectual y personal.

Por ello, toda empresa que decida invertir en formación dará a conocer a sus empleados el interés que tiene en ellos como personas, como trabajadores y como parte importante dentro de la organización.

El beneficio de la formación no es sólo para el trabajador, sino también para la empresa, que para ambos supone una inversión para enfrentar los retos del futuro.

Los beneficios son múltiples:

- ✓ Favorece la igualdad de oportunidades y la promoción personal y profesional.

- ✓ Permite al trabajador prepararse para la toma de decisiones y para la solución de problemas.
- ✓ Logra metas individuales.
- ✓ Eleva el nivel de satisfacción en el puesto de trabajo.
- ✓ Ayuda a la integración en la empresa.

Por todo esto, se ha pensado en un software que ayude a las empresas en esta tarea tan importante, de manera que simplifique la labor de gestionar sus actividades formativas. De ahí nace la idea de este proyecto.

1.2 Alcance del proyecto

Respecto al alcance del proyecto, comprende los siguientes puntos:

- ✓ Gestionar eficientemente los/as alumnos/as que cursarán las ofertas formativas, así como los/as instructores que las impartirán.
- ✓ Gestionar las distintas ediciones que tiene un curso.
- ✓ Sistema de preinscripciones y matriculación de alumnos, tanto preinscritos como no preinscritos.
- ✓ Seguimiento de la evaluación de los alumnos/as en las ediciones.
- ✓ Gestión de aulas para impartir las acciones formativas.

1.3 Requisitos del Sistema

1.3.1.1 Requisitos funcionales

A continuación se muestran los requisitos funcionales iniciales obtenidos de las (supuestas) entrevistas con los/as usuarios/as finales de la aplicación.

En el documento de análisis posterior se entrará a detallar en más profundidad y añadiendo el modelo de datos para cada parte o módulo de la aplicación.

Id	Descripción	Prioridad
RF1. Gestión de preinscripciones		
Actualización de datos		
RF1.1	Se permitirá preinscribir un/a alumno/a en una edición.	Alta
RF1.1.1	Para poder preinscribir un/a alumno/a en una edición, dicho/a alumno/a ha de estar dado/a de alta en el portal como alumno/a. Ver FR4.	Alta
RF1.1.1.1	Si el alumno/a a preinscribir no se ha dado de alta en el portal previamente, se facilitará su alta desde la zona de preinscripciones, permitiendo de esta manera, en un mimo paso, dar de alta al alumno/a y preinscribirlo en la edición.	Media
RF1.1.2	Existirá un buscador de alumnos/as no preinscritos/as en la edición, para facilitar la preinscripción.	Media
RF1.1.2.1	El buscador de alumnos/as no preinscritos/as en la edición, podrá filtrar por uno o varios de los siguientes campos: nombre, apellidos, categoría profesional y/o provincia.	Media
RF1.1.3	Se podrán preinscribir alumnos/as en la edición mientras esta no comience, dejando la fecha límite como la fecha de inicio de la edición.	Media
Consultas e Informes		
RF1.2	Listado de todos/as los preinscritos/as en una edición.	Alta
RF1.2.1	Se podrá imprimir el listado de alumnos/as preinscritos/as. No se permitirá imprimir listados vacíos.	Alta
RF1.3	Existirá una zona de gestión de alumnos/as, donde se permitirá listar y/o filtrar los/as alumnos/as dados/as de alta en el portal. RF6.3.	Media
RF2. Gestión de matrículas		
Actualización de datos		

RF2.1	Se permitirá matricular un/a alumno/a, esté o no preinscrito, en una edición.	Alta
RF2.1.1	Para poder matricular un/a alumno/a en una edición, dicho/a alumno/a ha de estar dado/a de alta en el portal como alumno/a. Ver RF6.1.	Alta
RF2.1.1.1	Si el alumno/a a matricular no se ha dado de alta en el portal previamente, se facilitará su alta desde la zona de matrículas de alumnos no preinscritos, permitiendo de esta manera, en un mimo paso, dar de alta al alumno/a y matricularlo en la edición.	Media
RF2.1.2	Para matricular un alumno/a que ya está preinscrito/a en la edición, se mostrará un listado de los alumnos/as preinscritos, y se checkeará uno o varios alumnos/as y se pulsará el botón matricular. Esta será la vía normal de matriculación.	Alta
RF2.1.3	Se podrá matricular un alumno/a no preinscrito/a en la edición. Para ello, existirá un buscador de alumnos/as no preinscritos/as ni matriculados/as. Se podrá checkear uno/a o varios/as alumnos/as y pulsar el botón matricular.	Alta
RF2.1.4	Se podrán matricular alumnos/as en la edición mientras esta no finalice, dejando la fecha límite a criterio del departamento de formación, según el curso y la dificultad de seguimiento del mismo.	Alta
RF2.1.4.1	No se podrán matricular alumnos/as en una edición, una vez que esta ya ha finalizado.	Alta
RF2.2	Se podrá dar de baja un alumno/a matriculado en una edición.	Alta
Consultas e Informes		
RF2.3	Listado de todos/as los matriculado/as en una edición.	Alta
RF2.4	Se podrá imprimir el listado de alumnos/as matriculados/as. No se permitirá imprimir listados vacíos.	Alta
RF3 Gestión de evaluaciones		
Actualización de datos		
RF3.1	Se podrán evaluar los/as alumnos/as de cada edición.	Alta
RF3.1.1	Para cada examen, se podrán seleccionar todos los alumnos/as de la edición (por defecto será así, pues es la opción más habitual), o sólo algunos/as de ellos/as.	Alta
RF3.2	Se permitirá acceder y modificar los datos de un examen seleccionado entre los resultados de las búsquedas y listados.	Media

RF3.2.1.2	Como excepción, el identificador no se podrá modificar bajo ninguna circunstancia.	Alta
RF3.3	Para cada alumno/a que realiza un examen, se podrá guardar la calificación que ha obtenido. La calificación podrá ser numérica y/o apto/no apto.	Alta
RF3.3.1	La calificación podrá ser numérica y/o apto-no apto.	Media
Consultas e Informes		
RF3.4	Listados de todos los exámenes, de una edición.	Alta
RF3.4.1	Se podrá imprimir el listado de alumnos/as que ha realizado un determinado examen, con su calificación. No se permitirá imprimir listados vacíos.	Alta
RF4. Gestión de cursos		
Actualización de datos		
RF4.1	Se permitirá introducir los datos de un nuevo curso.	Alta
RF4.1.1	Un curso, podrá disponer de uno o varios módulos.	Media
FR4.2	Se permitirá acceder y modificar los datos de un curso seleccionado entre los resultados de las búsquedas y listados.	Alta
RF4.2.1	Como excepción, el identificador no se podrá modificar bajo ninguna circunstancia.	Alta
Consultas e Informes		
RF4.3	Listado de todos los cursos impartidos por la empresa.	Media
RF4.3.1	Se podrá buscar un curso filtrando el listado anterior por nombre, tipo de contenido (sólo teórico, sólo práctico o teórico y práctico), modalidad (presencial, semipresencial, a distancia) y/o provincia.	Alta
RF4.3.1.1	Se permitirán búsquedas múltiples por varios campos.	Alta
RF4.3.1.2	No será necesario introducir todos los datos de búsqueda completos, el sistema deberá mostrar todas las coincidencias.	Alta
RF4.4	Se podrán imprimir todos los listados. No se permitirán imprimir listados vacíos.	Alta
RF5. Gestión de ediciones		
Actualización de datos		

RF5.1	Se permitirá introducir los datos de una nueva edición, a partir de un curso.	Alta
RF5.1.1	Por defecto, en el momento de seleccionar el curso a partir del cual se está creando la edición, se cogerán todos los datos del curso para la edición (módulos, número de horas, tipo de curso,...).	Alta
RF5.2	Una edición, podrá disponer de uno o varios instructores/as asociados.	Alta
RF5.2.1.1	Para asociar un/a instructor/a a una edición, deberá estar dado/a de alta en el portal. Ver RF5.	Alta
RF5.2.1.2	Si el instructor/a a asociar no se ha dado de alta en el portal previamente, se facilitará su alta desde la mima edición, permitiendo de esta manera, en un mimo paso, dar de alta al instructor/a y asignarlo a la edición.	Media
RF5.2.2	Cada instructor/a, tendrá una valoración, y se podrá actualizar el precio por hora y el tipo de instructor/a para esa edición (coordinador/a o profesor/a).	Alta
RF5.3	La edición contendrá los mismos módulos que el curso al que pertenece, y no se podrá añadir ni eliminar módulos.	Alta
RF5.3.1	Para cada módulo de la edición, se podrá asignar un/a instructor/a, de entre el listado de instructores/as de la edición.	Media
RF5.4	Cada edición tendrá un horario asignado.	Alta
RF5.5	Una edición se impartir en una o varias aulas. Ver RF8.	Alta
FR5.6	Se permitirá acceder y modificar los datos de una edición seleccionada entre los resultados de las búsquedas y listados.	Alta
RF5.6.1	Como excepción, el identificador no se podrá modificar bajo ninguna circunstancia.	Alta
Consultas e Informes		
RF5.7	Listado de todas las ediciones impartidas por la empresa.	Media
RF5.7.1	Se podrá buscar una edición filtrando el listado anterior por curso al que pertenece y/o rango de fechas de inicio de la edición.	Alta
RF5.7.1.1	Se permitirán búsquedas múltiples por varios campos.	Alta
RF5.7.1.2	No será necesario introducir todos los datos de búsqueda completos, el sistema deberá mostrar todas las coincidencias.	Alta

RF5.8	Se podrán imprimir todos los listados. No se permitirán imprimir listados vacíos.	Alta
RF5.9	Se permitirá listar todas las ediciones de una determinada provincia, en formato pdf.	Media
RF5.9.1	Los campos que aparecerán en el listado por provincia, en pdf son: identificador de la edición, nombre del curso al que pertenece, municipio en el que se imparte, lugar de impartición, fecha de presentación, fecha de inicio, instructor/a, número de alumnos/as matriculados/as, número de horas y modalidad.	Media
RF5.10	Para cada edición, se podrá imprimir una "Encuesta de satisfacción" en formato pdf, donde los/as alumnos/as podrán calificar la iluminación del aula, las instalaciones del centro, adecuación del material aportado para las clases teóricas y/o prácticas, si se han cumplido los horarios de entrada y salida y si los/as instructores/as solucionan adecuadamente las dudas que puedan surgir.	Baja
RF5.11	Para cada edición, se podrá imprimir una "Carta de presentación" en formato pdf, donde se especificará el curso al que pertenece la edición, las fechas de inicio y fin, el horario de impartición (incluyendo aula e instructor/a) y temario.	Baja
RF6. Gestión de alumnos/as		
Actualización de datos		
RF6.1	Se permitirá introducir los datos de un nuevo/a alumno/a.	Alta
RF6.2	Se permitirá acceder y modificar los datos de un/a alumno/a seleccionado entre los resultados de las búsquedas y listados.	Alta
RF6.2.1	Como excepción, el identificador no se podrá modificar bajo ninguna circunstancia.	Alta
Consultas e informes		
RF6.3	Listado con los/as alumnos/as de la aplicación.	Alta
RF6.3.1	Se facilitará la búsqueda de un alumno/a pudiendo filtrar el listado anterior por nombre, apellidos, categoría y/o provincia.	Alta
RF6.3.2	Se podrá acceder a los datos de un alumno/a seleccionado en el listado y modificar sus datos y su estado.	Alta
RF6.4	Se podrán imprimir todos los listados. No se permitirá imprimir listados vacíos.	Alta

RF7. Gestión de instructores/as		
Actualización de datos		
RF7.1	Se permitirá introducir los datos de un nuevo instructor/a.	Alta
RF7.1.1	Se podrá gestionar la información académica del instructor/a.	Media
RF7.1.1.1	Si se desea gestionar la información académica de un instructor/a, para cada título, se introducirá su denominación y, opcionalmente, las fechas de inicio y fin, número de horas, y una breve descripción.	Media
RF7.1.2	Se podrá gestionar documentación del instructor/a.	Baja
RF7.1.2.1	Si se desea gestionar documentación para el instructor/a, para cada documento se introducirá el fichero (documento en sí) y, opcionalmente, una breve descripción.	Media
RF7.2	Se permitirá acceder y modificar los datos de un/a instructor/a seleccionado entre los resultados de las búsquedas y listados.	Alta
RF7.2.1	Como excepción, el identificador no se podrá modificar bajo ninguna circunstancia.	Alta
Consultas e informes		
RF7.3	Listado con los/as instructores/as de la aplicación.	Alta
RF7.3.1	Se facilitará la búsqueda de un instructor/a pudiendo filtrar el listado anterior por nombre, apellidos, tipo (laboral/interno) y rango de precio por hora.	Alta
RF7.4	Se podrá acceder a los datos de un instructor/a seleccionado en el listado y modificar sus datos.	Alta
RF7.5	Se podrán imprimir todos los listados. No se permitirá imprimir listados vacíos.	Alta
RF7.6	Se podrá listar la información académica de un instructor/a.	Media
RF7.7	Se podrá listar la documentación de un instructor/a.	Media
RF8. Gestión de aulas		
Actualización de datos		
RF8.1	Se permitirá introducir los datos de una nueva aula.	Alta
RF8.2	Se permitirá acceder y modificar los datos de un aula seleccionado entre los resultados de las búsquedas y listados.	Alta
RF8.2.1	Como excepción, el identificador no se podrá modificar bajo	Alta

	ninguna circunstancia.	
Consultas e informes		
RF8.3	Listado de las aulas de formación.	Alta
RF8.3.1	Se facilitará la búsqueda de un aula pudiendo filtrar el listado anterior por nombre, provincia, municipio, población y rangos de capacidad, metros y valoración.	Media
RF8.4	Se podrá acceder a los datos de un aula seleccionada en el listado y modificar sus datos.	Alta
RF8.5	Se podrán imprimir todos los listados. No se permitirá imprimir listados vacíos.	Alta
RF9. Gestión de usuarios		
Usuarios de acceso a la aplicación		
RF9.1	Para acceder a la aplicación se deberán dar de alta usuarios y activarlos. Estos usuarios deberán autenticarse en el sistema antes de poder acceder.	Alta
RF9.1.1	Los usuarios podrán tener 2 estados: activos y no activos. Sólo los usuarios activos podrán acceder a la aplicación.	Alta
Actualización de datos		
RF9.2	Se permitirá introducir los datos de un nuevo usuario.	Alta
RF9.3	Se permitirá acceder y modificar los datos de un usuario seleccionado entre los resultados de las búsquedas y listados.	Alta
RF9.3.1	Como excepción, el identificador no se podrá modificar bajo ninguna circunstancia.	Alta
RF9.4	Si un usuario deja de serlo, se pasa a estado inactivo. El sistema deja de tratarle como usuario a todos los efectos. Se permitirá volver a activarlo como usuario en cualquier momento.	Alta
Consultas e informes		
RF9.5	Listado con los/as usuarios/as de la aplicación.	Alta
RF9.5.1	Se facilitará la búsqueda de un/a usuario/a pudiendo filtrar el listado anterior por nombre, apellidos, login y estado (activo/inactivo).	Alta
RF9.6	Se podrá acceder a los datos de un usuario seleccionado en el listado y modificar sus datos y su estado.	Alta

RF9.7	Se podrán imprimir todos los listados. No se permitirá imprimir listados vacíos.	Alta
RF9.8	Se podrán imprimir los datos guardados de los/as usuarios/as.	Alta
RF9.9	Para cada usuario/a, se podrá imprimir un informe en formato pdf, con sus datos.	Baja
RF9.9.1	En dicho informe, se especificará el nombre y apellidos, teléfono, móvil, nombre de usuario y contraseña de acceso a la aplicación, email, y su estado (Activo – Inactivo).	Baja

1.4 Tecnologías de desarrollo

Como lenguaje de programación, utilizaremos Java, pues ofrece múltiples ventajas sobre el resto de los lenguajes, por sus potentes características.

El lenguaje Java se creó con cinco objetivos principales:

- Debería usar la metodología de la programación orientada a objetos.
- Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Debería incluir por defecto soporte para trabajo en red.
- Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
- Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos.

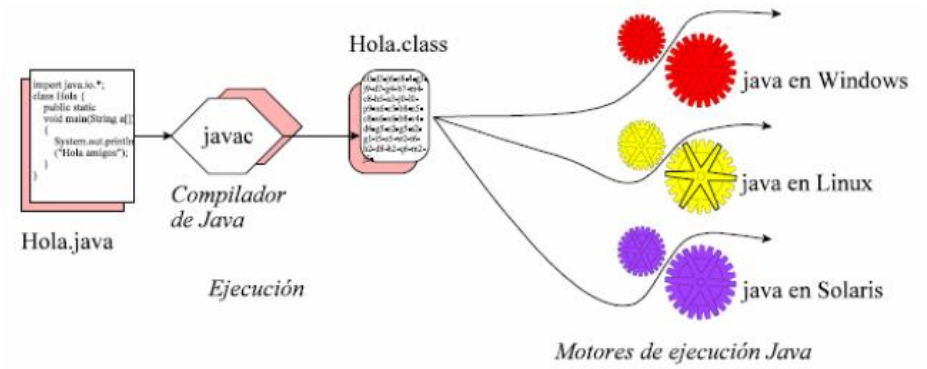


Ilustración 1 - Potencia de Java

Y basándonos en el lenguaje Java, utilizaremos la plataforma **JavaEE** (Java Platform, Enterprise Edition), pues se trata de un estándar para aplicaciones empresariales portables, escalares, robustas y seguras, con soporte para múltiples plataformas y sistemas operativos. Además, está avalado por múltiples empresas como SUM, IBM, ORACLE, etc. Nos ofrece competitividad y soluciones libres.

Por su parte, Struts: Apache Jakarta Struts (http://struts.apache.org) simplifica notablemente la implementación de una arquitectura según el patrón de diseño Modelo-Vista-Controlador el cual se utiliza ampliamente y es considerado de gran solidez. Es una plataforma cooperativa, eficaz y apropiada tanto para desarrolladores independientes como equipos grandes de desarrollo.

Struts permite que el desarrollador se concentre en el diseño de aplicaciones complejas como una serie simple de componentes del Modelo y de la vista intercomunicados por un control centralizado. Diseñando de esta manera puede obtenerse una aplicación más consistente y más fácil de mantener.

Por su parte, Spring es un framework que define la forma de desarrollar

aplicaciones J2EE, de contenedor liviano basado en la técnica Inversión de Control (IoC) y una implementación de desarrollo según el paradigma de Orientación a Aspectos (AOP). Simplifica enormemente la complejidad de los desarrollos bajo la plataforma J2EE, al promover el bajo acoplamiento a partir de la inyección de dependencias entre los objetos.

iBATIS Apache iBatis, está constituido por dos frameworks independientes que usaremos juntos: DAO y sqlMaps.

El primero, simplifica la implementación del patrón de diseño Direct Access Objects (DAO), mientras que el segundo, simplifica la persistencia de objetos en bases de datos relacionales. De este modo, el patrón DAO nos ayuda a organizar las tareas de persistencia (guardado, búsqueda, recuperación, borrado, etc.) de los objetos, y nos permite definir múltiples implementaciones para un mismo objeto mediante la definición de una interfaz.

El segundo, simplifica la tarea de conexión con la base de datos, y la forma de obtener los datos de ella, resumiendo la configuración en ficheros XML, con SQL ANSI o propietario, funcionando prácticamente con cualquier base de datos de driver JDBC.

Log4j es una biblioteca open source desarrollada en Java por la Apache Software Foundationque permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o “logs” (logging) en tiempo de ejecución y no en tiempo de compilación como es comúnmente realizado.

Con todo lo anterior, y como es de prever, para el diseño de la aplicación web, utilizaremos el patrón de diseño **Modelo-Vista-Controlador**, puesto que la lógica de interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Se trata de realizar un diseño que desacopla la vista del modelo, con la finalidad de mejorar la reusabilidad.

Elementos del patrón:

- Modelo: datos y reglas de negocio.
- Vista: muestra la información del modelo al usuario/a.
- Controlador: gestiona las entradas del usuario/a.

Veamos un esquema:

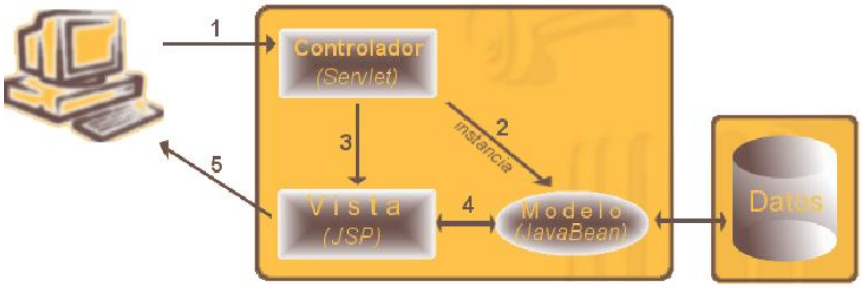
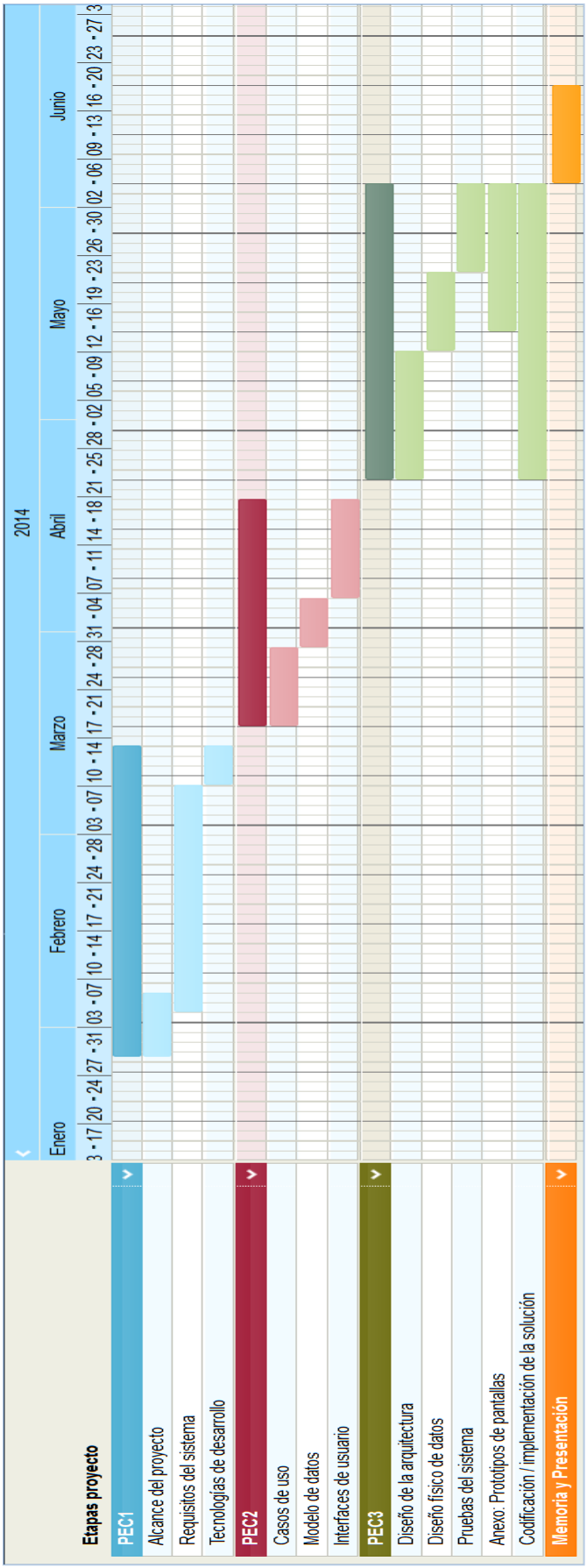


Ilustración 2 - Patrón de diseño MVC

1.5 Planificación



2 Análisis del Sistema

2.1 Especificación funcional del sistema

2.2 Introducción

La especificación funcional se divide en nueve subsistemas, como se puede ver en el diagrama siguiente:

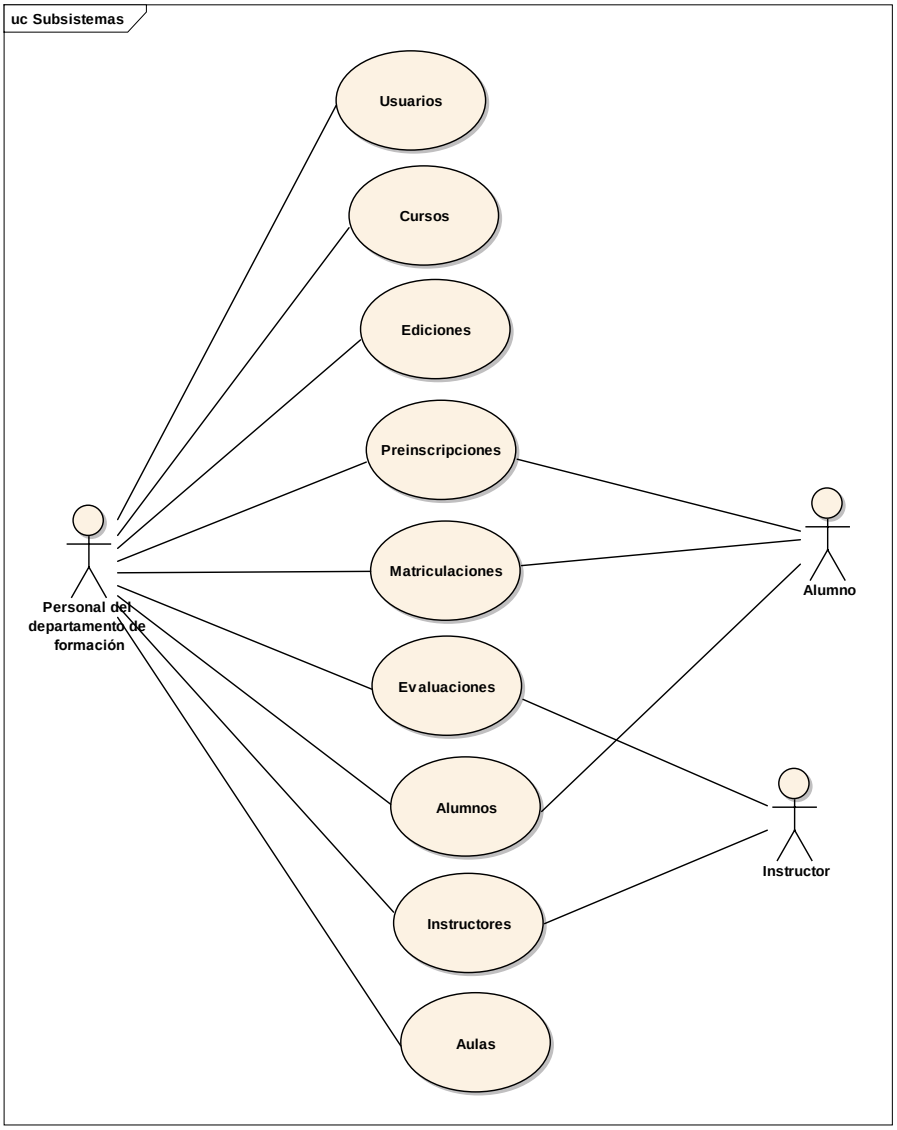


Ilustración 3 - Diagrama general del sistema

Veamos en qué consiste cada subsistema:

- **Subsistema "Usuarios":** se encarga de gestionar los usuarios/as que tendrán acceso a la aplicación. Serán personal del departamento de formación de la empresa.
- **Subsistema "Cursos":** gestiona los cursos que oferta la empresa a sus empleados. Se guardan los datos generales del curso para, posteriormente, realizar ediciones de los mismos a partir de estos datos comunes.
- **Subsistema "Ediciones":** se hace cargo de la gestión de ediciones de un curso. Estas, serán impartidas por instructores/as, que habitualmente, será personal de la empresa (ocasionalmente, podrá contratarse a personal ajeno a la empresa). Se establecerá un horario para cada edición, y una o varias aulas de impartición.
- **Subsistema "Preinscripciones":** se trata de la gestión de preinscripciones de los alumnos/as a una edición de un curso. Habitualmente, será el proceso previo a la matriculación del mismo.
- **Subsistema "Matriculaciones":** se encarga de gestionar las matriculaciones de los alumnos/as a una edición de un curso. Estas matriculaciones podrán ser de alumnos previamente preinscritos en la edición, o que no se han preinscrito.
- **Subsistema "Evaluaciones":** gestiona las evaluaciones de los alumnos/as para una determinada edición. Se trata de pruebas, actividades, o exámenes que se califican, con el fin de evaluar el aprendizaje de cada alumno/a.
- **Subsistema "Alumnos":** encargado de gestionar los alumnos/as que recibirán la formación. Serán siempre personal de la empresa.
- **Subsistema "Instructores":** encargado de gestionar los instructores/as que impartirán la formación. Por regla general, serán personal de la empresa más experimentado en la materia a impartir. Ocasionalmente, podrá ser personal ajeno a la empresa.
- **Subsistema "Aulas":** encargado de gestionar las aulas en las que se impartirá la formación.

A continuación, vamos a ir viendo los casos de uso para cada subsistema. En la descripción de los mismos, seguiremos unas reglas genéricas, común para todos ellos:

- *[Manual]:* indica que la acción se requiere para el caso de uso, pero es ajena al sistema en el sentido de que no se interactúa con él.
- *[Nota]:* aclaración que conviene tener en cuenta, pero que no interviene en el sistema.
- *Lista numerada:* las acciones se realizan según el orden indicado en dicha numeración.
- *Lista alfabética:* indica flujos alternativos. Es decir, se realiza una u otra opción (si se realiza la opción a), no se realiza la b) ni la c),…).

2.3 Modelos de casos de uso

2.3.1 Modelo del subsistema “Usuarios”

2.3.1.1 Diagramas de casos de uso

Los actores participantes son los siguientes:

- *Personal del departamento de formación:* incluye a todos los usuarios finales que utilizarán la aplicación.

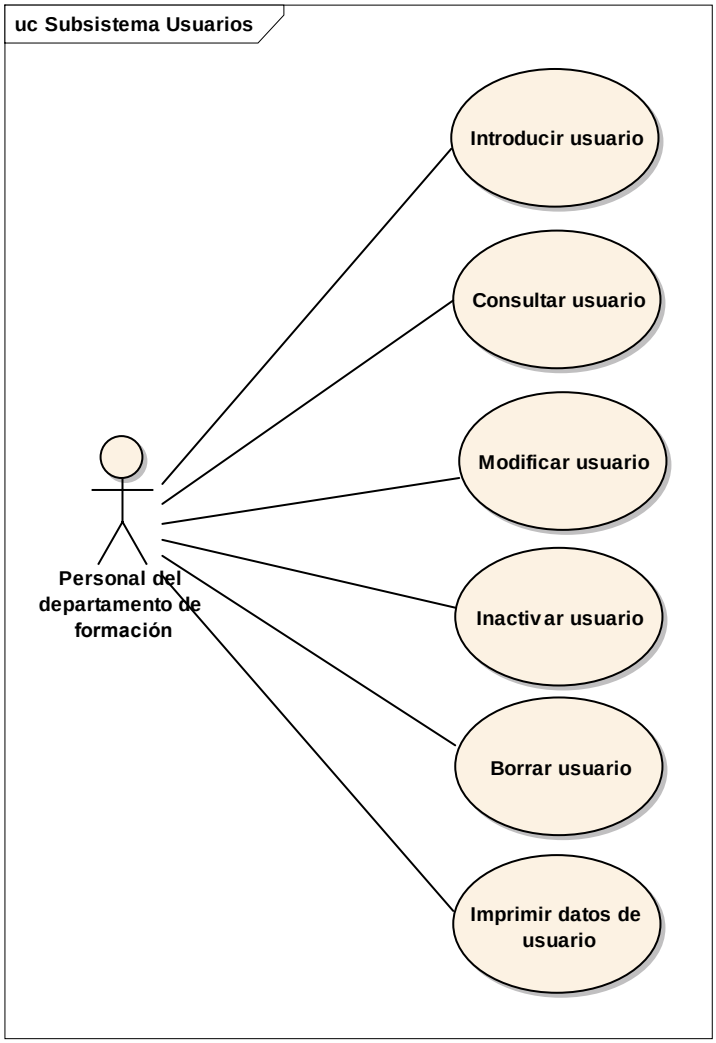


Ilustración 4 - Modelo del subsistema "Usuarios"

2.3.1.2 Descripción de casos de uso

Caso de uso: "Introducir datos de usuario"

Precondiciones: -
Pos condiciones: el usuario/a queda dado de alta en el sistema. Por lo tanto, podrá comenzar a utilizar la aplicación.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. El personal de formación introduce el nombre y apellidos del usuario/a, login, contraseña, confirmación de la contraseña, teléfono, móvil, email y estado (activo o inactivo).
2. El sistema comprueba que los datos sean correctos (contraseña y confirmación de la contraseña iguales, email con formato válido y todos los datos obligatorios rellenos).
   a. Si todo es correcto, el sistema crea el usuario/a y muestra el mensaje de éxito.
   b. Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.
3. [Nota] Hay que tener en cuenta que, si se crea un usuario/a con estado inactivo, no podrá utilizar la aplicación. Sólo tienen acceso a la aplicación los usuarios/as en estado activo.

Caso de uso: "Consultar datos de usuario"

Precondiciones: los datos del usuario a consultar, deben estar guardados en el sistema.
Pos condiciones: -
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Mediante el buscador de usuarios, se busca el usuario que se desea consultar, pudiendo filtrar por el nombre, apellidos, login y/o estado (activo/inactivo).
2. Una vez localizado el usuario que se desea consultar, ha de pulsarse sobre el nombre del mismo.
3. El sistema nos mostrará en una pantalla idéntica a la de alta de usuario, con los datos del mismo.

Caso de uso: "Modificar datos de usuario"

Precondiciones: el usuario a modificar deben estar dado de alta en el sistema.
Pos condiciones: los datos del usuario/a quedan actualizados.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar el usuario que se desea modificar (ver caso de uso 2.2.1.2.2).
2. Modificar los datos que se desee.
3. El sistema comprueba que los datos sean correctos (de la misma manera que en el alta del usuario).
   a) Si todo es correcto, el sistema guarda los cambios en el usuario y muestra el mensaje de éxito.
   b) Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

Caso de uso: "Inactivar usuario"

Precondiciones: el usuario a inactivar debe estar dado de alta en el sistema.
Pos condiciones: el usuario deja de serlo a todos los efectos, pero se podrá consultar sus datos o volver a activarlo.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Modificar el usuario que se desea inactivar (ver caso de uso 2.2.1.2.3), pero modificando únicamente el check de estado de usuario (quedando este desmarcado).
2. El sistema comprueba que los datos sean correctos (de la misma manera que en el caso de uso 2.2.1.2.3) y procederá de la misma manera.

Caso de uso: "Borrar usuario"

Precondiciones: el usuario a borrar, debe estar dado de alta en el sistema.
Pos condiciones: el usuario deja de serlo a todos los efectos. No se podrá recuperar.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Mediante el buscador de usuarios, se busca el usuario que se desea eliminar, pudiendo filtrar por el nombre, apellidos, login y/o estado (activo/inactivo).
2. Una vez localizado el usuario a borrar, se indica al sistema la intención de eliminarlo definitivamente.
3. El sistema nos mostrará el siguiente mensaje de confirmación del borrado: "Los datos borrados no se pueden recuperar. ¿Proceder con el borrado?".
    a) Si se pulsa aceptar, el sistema lo elimina definitivamente, y vuelve a mostrar el listado anterior resultante de la búsqueda con los datos actuales (sin el usuario borrado).
    b) Si se pulsa cancelar, el sistema no realiza ninguna acción.

Caso de uso: "Imprimir datos de usuario"

Precondiciones: el usuario que se desea imprimir, debe de estar dado de alta previamente en el sistema.
Pos condiciones: se imprimen los datos del usuario/a en formato pdf.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar el usuario para el que se desea imprimir sus datos (ver caso de uso 2.2.1.2.2).
2. Pulsando sobre el enlace "Obtener informe" se obtendrá un archivo en formato pdf con los datos del usuario.

2.3.2 Modelo del subsistema “Curso”

2.3.2.1 Diagramas de casos de uso

Los actores participantes son los siguientes:

- *Personal del departamento de formación:* incluye a todos los usuarios finales que utilizarán la aplicación.

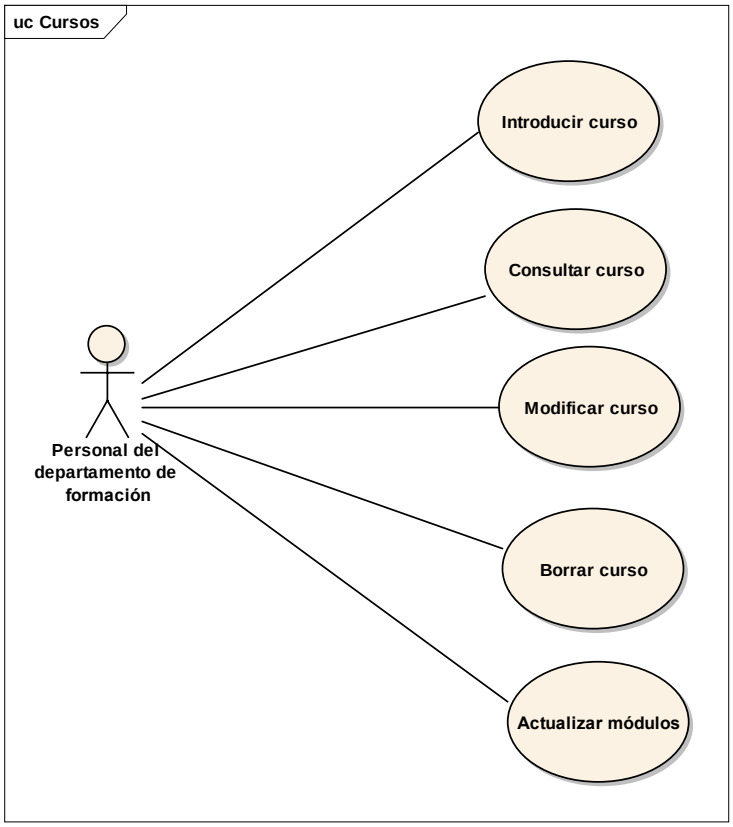


Ilustración 5 - Modelo del subsistema "Curso"

2.3.2.2 Descripción de casos de uso

Caso de uso: "Introducir datos de curso"

Precondiciones: -
Pos condiciones: el curso queda dado de alta en el sistema.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. El personal de formación introduce el nombre del curso, información, temario, tipo de contenido (teórico, práctico o mixto), número de horas teóricas y/o prácticas, provincia, modalidad (presencial, semipresencial o a distancia) y, opcionalmente, observaciones.
2. El sistema debe comprobar que los datos sean correctos (número de horas teóricas y/o prácticas sea coherente con el contenido –teórico, práctico, mixto - de la edición y todos los datos obligatorios están rellenos).
   a) Si todo es correcto, el sistema crea el curso y muestra el mensaje de éxito.
   b) Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

Caso de uso: "Consultar datos de curso"

Precondiciones: los datos del curso a consultar, deben estar guardados en el sistema.
Pos condiciones: -
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Mediante el buscador de cursos, se busca el curso que se desea consultar, pudiendo filtrar por el nombre, tipo de contenido, modalidad, provincia y rangos de horas teóricas y/o prácticas del curso.
2. Una vez localizado el curso que se desea consultar, ha de pulsarse sobre el nombre del mismo.
3. El sistema nos mostrará en una pantalla idéntica a la de alta de curso, los datos del mismo.

Caso de uso: "Modificar datos de curso"

Precondiciones: los datos del curso a modificar, deben estar guardados en el sistema.
Pos condiciones: los datos del curso quedan actualizados.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar el curso que se desea modificar (ver caso de uso 2.2.2.2.2).
2. Modificar los datos que se desee.
3. El sistema comprueba que los datos sean correctos (de la misma manera que en el alta del curso).
   a) Si todo es correcto, el sistema guarda los cambios en el curso y muestra el mensaje de éxito.
   b) Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

Caso de uso: "Borrar curso"

Precondiciones: los datos del curso a borrar, deben estar guardados en el sistema.
Pos condiciones: el curso el eliminado del sistema.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Excepciones: si ya se han creado ediciones del curso.
Descripción:

1. Mediante el buscador de cursos, se busca el curso que se desea eliminar, pudiendo filtrar por el nombre, tipo de contenido, modalidad, provincia y rangos de horas teóricas y/o prácticas del curso.
2. Una vez localizado el curso a borrar, se comunica al sistema la intención de eliminarlo definitivamente.
3. El sistema nos mostrará el siguiente mensaje de confirmación del borrado: "Los datos borrados no se pueden recuperar. ¿Proceder con el borrado?".
   a) Si se pulsa aceptar, el sistema comprobará que no se hayan creado ediciones del mismo.
      a. Si no hay ediciones del curso, lo elimina definitivamente, y vuelve a mostrar el listado anterior resultante de la búsqueda con los datos actuales (sin el curso borrado).
      b. En caso contrario, el sistema muestra un mensaje indicando que ya hay ediciones para ese curso, por lo que no se puede eliminar.
   b) Si se pulsa cancelar, el sistema no realiza ninguna acción.

Caso de uso: "Actualizar módulos"

Precondiciones: el curso al que se desea actualizar los módulos de los que consta, debe estar guardado en el sistema.
Pos condiciones: los módulos del curso, quedan actualizados.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Excepciones: -
Descripción:

1. Mediante el buscador de cursos, se busca el curso al que se desea actualizar sus módulos, pudiendo filtrar por el nombre, tipo de contenido, modalidad, provincia y rangos de horas teóricas y/o prácticas del curso.
2. Una vez situado en el curso, dirigiéndose a la zona de módulos del mismo, el sistema nos muestra los módulos de los que consta el curso.
   a) Si se desea modificar los datos de un módulo, se pulsa en el nombre del mismo. De esta manera, se podrá cambiar su nombre, o el tipo de contenido en el caso de que el curso sea mixto (teoría y práctica).
   b) Si se desea eliminar un determinado módulo, el sistema nos mostrará el siguiente mensaje de confirmación del borrado: "Los datos borrados no se pueden recuperar. ¿Proceder con el borrado?".
      a. Si se pulsa aceptar, lo elimina definitivamente, y vuelve a mostrar el listado anterior con los datos actuales (sin el módulo borrado).
      b. Si se pulsa cancelar, el sistema no realiza ninguna acción.
   c) Si se desea añadir un nuevo módulo, se pulsa en la pestaña "Nuevo". De esta manera, se da la opción de rellenar los datos del módulo (nombre y tipo de contenido en el caso de que el curso sea mixto). Pulsando el botón "Añadir", el sistema comprueba que todos los datos estén rellenos.
      a. Si es así, el sistema muestra un mensaje de éxito, y crea el módulo.
      b. En el caso contrario, muestra un mensaje de error, indicando qué campo no se ha rellenado.

2.3.3 Modelo del subsistema “Edición”

2.3.3.1 Diagramas de casos de uso

Los actores participantes son los siguientes:

- *Personal del departamento de formación:* incluye a todos los usuarios finales que utilizarán la aplicación.
- *Alumnos:* incluye a todos los empleados de la empresa que quieran recibir alguna formación de las ofertadas.
- *Instructores:* personal encargado de impartir la formación.

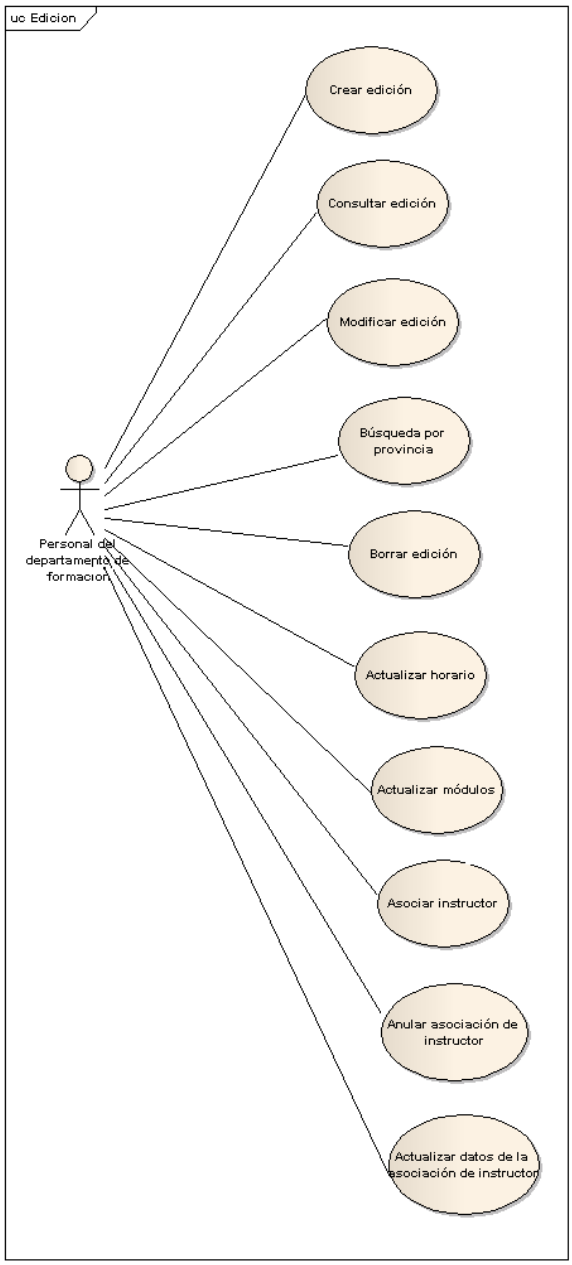


Ilustración 6 - Modelo del subsistema "Edición"

2.3.3.2 Descripción de casos de uso

Caso de uso: "Crear edición"

Precondiciones: debe de estar dado de alta el curso del que se desea crear una edición.
Pos condiciones: la edición es creada en el sistema.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Se selecciona el curso del cuál se pretende crear una edición.
2. El sistema rellenará automáticamente el número de horas teóricas y/o prácticas del curso.
   2.1. Estas horas, se pueden modificar para la edición que se está creando.
3. El sistema indicará, a partir del curso seleccionado, si el contenido de la edición es teórico, práctico o mixto.
4. El usuario/a rellena la fecha de inicio y fin de la edición y, opcionalmente, la fecha de presentación de la misma. Hay que tener en cuenta que, la fecha de presentación, podrá ser dentro de la impartición de la edición o anterior al inicio de la misma. Opcionalmente, se podrá introducir las observaciones que se consideren oportunas.
5. El sistema comprueba que los datos sean correctos (fecha de fin de edición posterior a la de inicio, fecha de presentación anterior al inicio de la edición o durante su impartición, y que el número de horas teóricas y/o prácticas sea coherente con el contenido de la edición).
   a) Si todo es correcto, el sistema crea la edición y muestra el mensaje de éxito.
   b) Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

Caso de uso: "Consultar edición"

Precondiciones: los datos de la edición a consultar, deben estar guardados en el sistema.
Pos condiciones: -
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Mediante el buscador de ediciones, se busca la edición que se desea consultar, pudiendo filtrar por el nombre de curso al que pertenece, rangos de fechas de inicio de la edición y rangos de horas teóricas y/o prácticas de la edición.
2. Una vez localizada la edición que se desea consultar, ha de pulsarse sobre el nombre de la misma.
3. El sistema nos mostrará en una pantalla idéntica a la de alta de la edición, los datos de la misma.

Caso de uso: "Modificar edición"

Precondiciones: los datos de la edición a modificar, deben estar guardados en el sistema.
Pos condiciones: la edición queda actualizada.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar la edición que se desea modificar (ver caso de uso 2.2.3.2.2).
2. Modificar los datos que se desee.
3. El sistema comprueba que los datos sean correctos (de la misma manera que en el alta de ediciones).
   a) Si todo es correcto, el sistema guarda los cambios en la edición y muestra el mensaje de éxito.
   b) Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

Caso de uso: "Búsqueda por provincia"

Precondiciones: -
Pos condiciones: se genera un fichero en formato pdf con el listado de ediciones que se imparte en la provincia seleccionada.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Seleccionar la provincia para la que se desea consultar las ediciones que no han finalizado.
2. El sistema, mostrará un enlace hacia un archivo en formato pdf. Al pulsar sobre él, se dará la opción al usuario/a de guardarlo, o abrirlo directamente.
   2.1. El archivo pdf contendrá el listado de ediciones que se están impartiendo en la provincia seleccionada, o que aún no han comenzado a impartirse.

Caso de uso: "Borrar edición"

Precondiciones: los datos de la edición a borrar, deben estar guardados en el sistema.
Pos condiciones: la edición es eliminada del sistema.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Mediante el buscador de ediciones, se busca la edición que se desea eliminar, pudiendo filtrar por el nombre de curso al que pertenece, rangos de fechas de inicio de la edición y rangos de horas teóricas y/o prácticas de la edición.
2. Una vez localizada la edición que se desea borrar, se indica al sistema que se desea eliminar definitivamente.
3. El sistema nos mostrará el siguiente mensaje de confirmación del borrado: "Los datos borrados no se pueden recuperar. ¿Proceder con el borrado?".
   a. Si se pulsa aceptar, el sistema elimina definitivamente la edición, y vuelve a mostrar el listado anterior resultante de la búsqueda con los datos actuales (sin la edición borrada).
   b. Si se pulsa cancelar, el sistema no realiza ninguna acción.

Caso de uso: "Actualizar horario"

Precondiciones: la edición a la que se desea actualizar el horario, debe estar dada de alta en el sistema.
Pos condiciones: la edición tiene un horario actualizado.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar la edición para la que se desea actualizar el horario (ver caso de uso 2.2.3.2.2), e ir a la zona de horario.
2. El sistema nos mostrará las tandas ya creadas del horario, en el caso de que hubiera. También nos dará la opción de borrar una tanda o introducir una nueva.
   a) Si se quiere borrar una tanda del horario, se indica al sistema. Este mostrará un mensaje de confirmación de borrado.
   b) Si se desea añadir una nueva tanda al horario, ha de introducirse fecha de inicio y fecha de fin, los días lectivos que tendrá (días de la semana), la hora de inicio y de fin, el aula en el que impartirá, y si se impartirá teoría o práctica (en el caso de que el curso sea de contenido mixto).
      2.b.1. El sistema comprobará que todos los datos sean válidos y coherentes, es decir, que no se seleccionen fechas fuera del rango de impartición dela edición, que la fecha de inicio de la tanda sea anterior a la de fin (ídem con las horas), que el aula no esté ya ocupada, y que el tipo de contenido sea coherente con la edición (por ejemplo, una edición teórica, no puede contener tandas con contenido práctico).

a. Si todo es correcto, el sistema guarda la tanda del horario y muestra el mensaje de éxito.
b. Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

Caso de uso: "Actualizar módulos"

Precondiciones: la edición para la que se desea actualizar los módulos, debe estar guardada en el sistema. Los módulos deben de estar creados en el curso al que pertenece la edición.
Pos condiciones: las fechas de impartición de los módulos, así como los instructores/as que lo imparten, quedan actualizados.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar la edición para la que se desea actualizar sus módulos (ver caso de uso 2.2.3.2.2), y dirigirse a la zona de módulos de la edición.
2. El sistema nos mostrará los módulos de la edición, en el caso de que tenga. Hay que tener en cuenta que, estos módulos son los del curso al que pertenece la edición.
3. Se da la opción de actualizar el instructor/a asociado a cada módulo (de entre los instructores de la edición).
4. El sistema mostrará el mensaje de éxito, y guardará los cambios realizados.

Caso de uso: "Asociar instructor"

Precondiciones: la edición para la que se desea asociar instructores/as, debe estar guardada en el sistema. El instructor/a que se desea asociar, también tiene que estar dado previamente de alta en el sistema.
Pos condiciones: el instructor/a queda asociado/a a la edición.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar la edición para la que se desea asignar instructores (ver caso de uso 2.2.3.2.2) y dirigirse a la zona de instructores/as asociados/as.
2. El sistema nos mostrará los instructores ya asociados a la edición, en el caso de que hubiera.
3. El usuario puede marcar uno o varios instructores/as para asignar a la edición.
4. El sistema asigna el/los instructores/as señalados, a la edición, y muestra un mensaje de éxito.

Caso de uso: "Anular asociación de instructor"

Precondiciones: la edición para la que se desea anular asociación instructores/as, debe estar guardada en el sistema. El instructor/a que se desea anular la asociación, también tiene que estar dado previamente de alta en el sistema, y asociado a dicha edición.
Pos condiciones: el instructor/a ya no está asociado/a a la edición.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Excepciones: si el instructor/a no estaba previamente asociado a la edición.
Descripción:

1. Consultar la edición para la que se desea anular la asignación de instructor (ver caso de uso 2.2.3.2.2) y dirigirse a la zona de instructores/as asociados/as.
2. El sistema nos mostrará los/as instructores/as asignados/as a la edición, en el caso de que tenga.
3. Una vez localizado el/la instructor/a que se desasignar de la edición, ha de indicarlo al sistema.
4. El sistema nos mostrará el siguiente mensaje de confirmación del borrado: "Los datos borrados no se pueden recuperar. ¿Proceder con el borrado?".
   a) Si se pulsa aceptar, el sistema elimina la asignación de instructor en la edición, y

vuelve a mostrar el listado actualizado de instructores/as de la edición.

b) Si se pulsa cancelar, el sistema no realiza ninguna acción.

Caso de uso: "Actualizar datos de la asociación de instructor"

Precondiciones: la edición para la que se desea actualizar los datos de un instructor/a asociado/a, debe estar guardada en el sistema. El instructor/a que se desea actualizar la asociación, también tiene que estar dado previamente de alta en el sistema, y asociado a dicha edición.

Pos condiciones: los datos del instructor/a para esa edición (precio de dietas, precio del kilometraje, si tiene o no coche de empresa y el tipo de instructor/a dentro de la edición, es decir, si es coordinador/a o profesor/a), quedan actualizados.

Quien lo comienza: personal del departamento de formación de la empresa.

Quien lo finaliza: personal del departamento de formación de la empresa.

Excepciones: si el instructor/a no estaba previamente asociado a la edición.

Descripción:

1. Consultar la edición para la que se desea actualizar los datos de sus instructores asociados (ver caso de uso 2.2.3.2.2) y dirigirse a la zona de instructores de la edición.
2. El sistema nos mostrará los instructores ya asociados a la edición, en el caso de que hubiera.
3. El usuario/a puede actualizar el precio por hora y el rol que jugará en la misma (coordinador o profesor, entendiendo que, el coordinador también puede impartir clase).
   3.1. Puede actualizar a la vez a uno o más instructores.
   3.2. El sistema comprueba si los datos introducidos son válidos:
      a) Si el sistema encuentra algún dato no válido, muestra un mensaje al usuario/a con los errores encontrados.
      b) Si el sistema no encuentra datos inválidos, guarda los datos actualizados de los instructores para esa edición.

2.3.4 Modelo del subsistema "Preinscripciones"

2.3.4.1 Diagramas de casos de uso

Los actores participantes son los siguientes:

- *Personal del departamento de formación:* incluye a todos los usuarios finales que utilizarán la aplicación.
- *Alumnos:* incluye a todos los empleados de la empresa que quieran recibir alguna formación de las ofertadas.

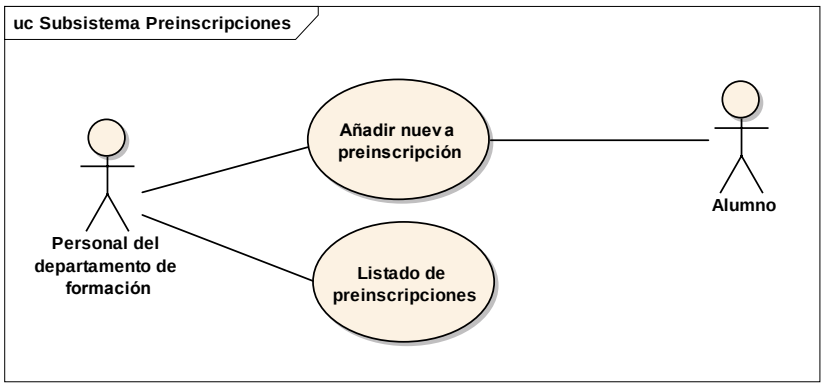


Ilustración 7- Modelo del subsistema "Preinscripciones"

2.3.4.2 Descripción de casos de uso

Caso de uso: "Añadir nueva preinscripción"
Precondiciones: La edición en la que se desea preinscribir el alumno/a, también ha de estar previamente dada de alta en el sistema. **Pos condiciones:** el alumno/a queda preinscrito en el sistema. **Quien lo comienza:** personal de la empresa que desee recibir la formación, que jugará el papel de alumno/a. **Quien lo finaliza:** personal del departamento de formación de la empresa. **Descripción:** 1. [Manual] Personal de la empresa se pone en contacto con el departamento de formación, mostrando su interés en recibir una determinada formación de las que se ofertan. 2. Consultar la edición para en la que se desea preinscribir el/la alumno/a (ver caso de uso 2.2.3.2.2) y dirigirse a la zona de preinscripciones. 3. El sistema nos mostrará los alumnos/as ya preinscritos/as, y dará la opción de buscar los aún no preinscritos, filtrando por nombre, apellidos, provincia y/o categoría. a) Si el alumno/a a preinscribir no se encuentra en el listado anterior, porque no está registrado en el sistema, se da la opción de introducir el alumno desde aquí. 3.a.1. Rellenando todos sus datos, este quedará automáticamente preinscrito en el sistema. b) Si el alumno/a a preinscribir se encuentra en el listado resultante de la búsqueda, se marca, y este queda preinscrito en el sistema. 3.b.1. Se pueden marcar uno o varios alumnos/a para preinscribir a la vez.

Casos de uso: "Listado de preinscripciones"
Precondiciones: La edición de la que se desea consultar las preinscripciones realizadas, ha de estar dada de alta en el sistema. **Pos condiciones:** - **Quien lo comienza:** personal del departamento de formación de la empresa. **Quien lo finaliza:** personal del departamento de formación de la empresa. **Descripción:** 1. Consultar la edición para en la que se desea ver el los/as alumnos/as preinscritos (ver caso de uso 2.2.3.2.2) y dirigirse a la zona de preinscripciones. 2. El sistema nos mostrará los alumnos/as ya preinscritos/as en la edición, pudiendo exportar el listado de los mismos a un fichero en formato pdf o Excel.

2.3.4.3 Diagramas de casos de uso

Los actores participantes son los siguientes:

- *Personal del departamento de formación:* incluye a todos los usuarios finales que utilizarán la aplicación.
- *Alumnos:* incluye a todos los empleados de la empresa que quieran recibir alguna formación de las ofertadas.

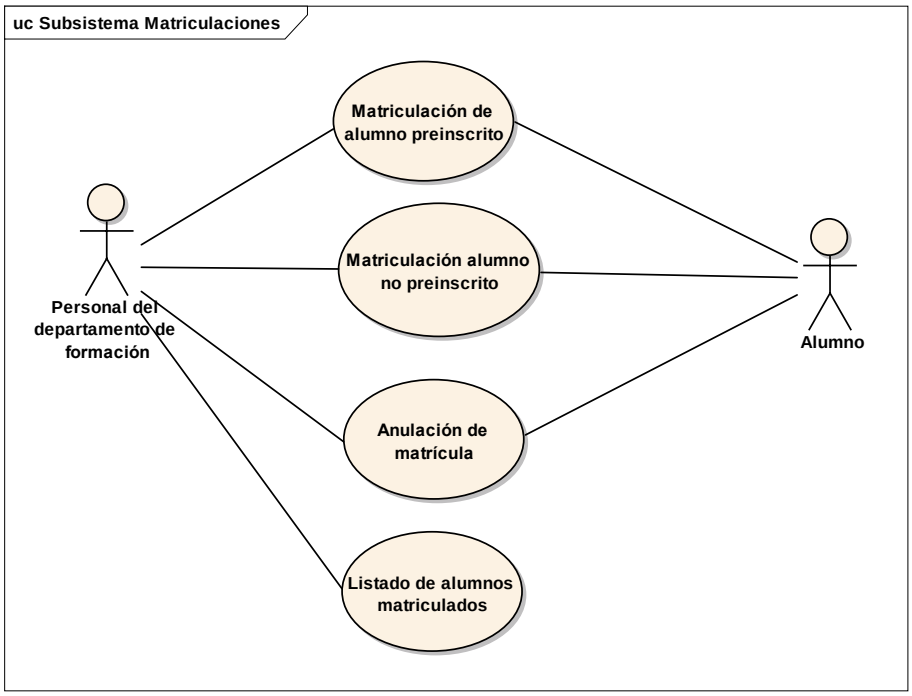


Ilustración 8 - Modelo del subsistema "Matriculaciones"

2.3.4.4 Descripción de casos de uso

Casos de uso: "Matriculación de alumno preinscrito"

Precondiciones: El alumno/a que se desea matricular en una edición, debe de estar preinscrito/a previamente en la edición.
Pos condiciones: el alumno/a queda matriculado en la edición. Ya es formalmente, un alumno/a de la edición.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Consultar la edición para en la que se desea matricular el/la alumno/a (ver caso de uso 2.2.3.2.2) y dirigirse a la zona de matriculaciones.
2. El sistema nos muestra un listado con los alumnos/as ya matriculados/as y otro con los alumnos/as preinscritos/as en la edición, pero que aún no se han matriculado.
   2.1. Seleccionando uno o varios alumnos/as del listado de preinscritos/as, estos quedan matriculados/as en la edición.

Caso de uso: "Matriculación de alumno no preinscrito"

Precondiciones: El alumno/a que se desea matricular en una edición, debe de estar dado de alta previamente en el sistema.
Pos condiciones: el alumno/a queda matriculado en la edición. Ya es formalmente, un alumno/a de la edición.
Quien lo comienza: personal de la empresa que desee recibir la formación, que jugará el papel de alumno/a.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. [Manual] Personal de la empresa se pone en contacto con el departamento de formación, mostrando su interés en recibir una determinada formación de las que se ofertan.
2. El personal del departamento de desarrollo consulta la edición para en la que se desea matricular el/la alumno/a (ver caso de uso 2.2.3.2.2) y se dirige a la zona de matriculaciones de alumnos/as no preinscritos/as.
3. El sistema nos muestra los alumnos/as ya matriculados/as, y dará la opción de buscar los aún no matriculados ni preinscritos, filtrando por nombre, apellidos, provincia y/o categoría.
   a) Si el alumno/a a matricular no se encuentra en el listado anterior, porque no está registrado en el sistema, se da la opción de introducir el alumno desde aquí.
      3.a.1. Rellenando todos sus datos, este quedará automáticamente matriculado en el sistema.
   b) Si el alumno/a a matricular se encuentra en el listado resultante de la búsqueda, se marca. De esta manera, queda matriculado/a en el sistema.
      3.b.1. Se da la opción de marcar uno o varios alumnos/as a la vez, para matricularlos en la edición.

Caso de uso: "Anulación de matrícula"
Precondiciones: El alumno/a que se desea dar de baja, debe de estar matriculado/a previamente en la edición. **Pos condiciones:** el alumno/a se da de baja en la edición. **Quien lo comienza:** un alumno/a de la edición. **Quien lo finaliza:** personal del departamento de formación de la empresa. **Descripción:** 1. [Manual] Uno o varios alumnos de una edición se ponen en contacto con el departamento de formación, mostrando su intención de causar baja en una determinada formación en la que están matriculados. 2. El personal del departamento de desarrollo consulta la edición para en la que se causar baja (ver caso de uso 2.2.3.2.2) y se dirige a la zona de cambio de estado de la matrícula. 3. El sistema nos muestra los alumnos/as matriculados/as, dando la opción de dar de baja a uno o varios alumnos/as a la vez. 4. Los estados de los alumnos en la edición quedan actualizados.

Caso de uso: "Listado de alumnos matriculados"
Precondiciones: - **Pos condiciones:** - **Quien lo comienza:** personal del departamento de formación de la empresa. **Quien lo finaliza:** personal del departamento de formación de la empresa. **Descripción:** 1. El personal del departamento de desarrollo consulta la edición para en la que desea obtener el listado de alumnos matriculados (ver caso de uso 2.2.3.2.2) y se dirige a la zona de matriculaciones. 2. El sistema nos muestra los alumnos/as matriculados/as, mostrando el color verde, los que están matriculados, y en color rojo lo que, estando matriculados, han causado baja. 3. Si se pulsa sobre los enlaces "Excel" o "pdf", se da la opción de importar el listado de alumnos/as matriculados a formato Excel o pdf respectivamente.

2.3.5 Modelo del subsistema "Alumnos"

2.3.5.1 Diagramas de casos de uso

Los actores participantes son los siguientes:

- *Personal del departamento de formación:* incluye a todos los usuarios finales que utilizarán la aplicación.
- *Alumnos:* incluye a todos los empleados de la empresa que quieran recibir alguna formación de las ofertadas.

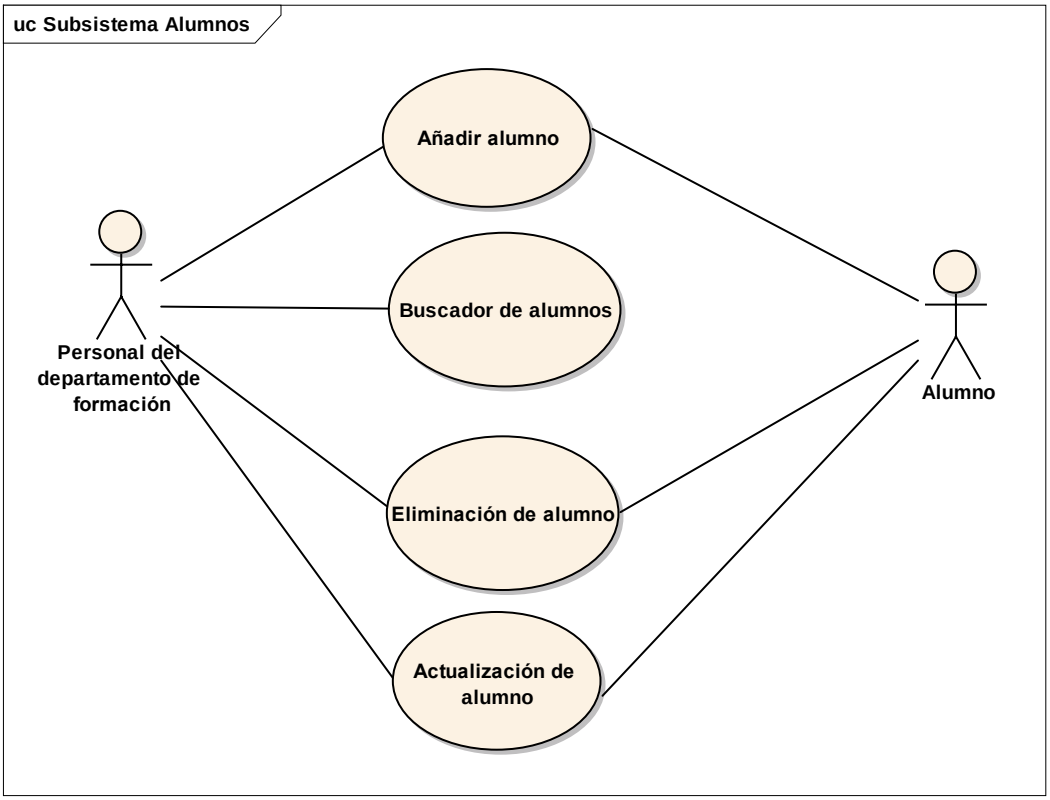


Ilustración 9 - Modelo del subsistema "Alumnos"

2.3.5.2 Descripción de casos de uso

Caso de uso: "Añadir alumno"
Precondiciones: - **Pos condiciones:** el alumno/a queda dado de alta en el sistema. Por lo tanto, ya se podrá preinscribir o matricular en las ediciones. **Quien lo comienza:** personal de la empresa que desea recibir formación, o el departamento de formación de la empresa. **Quien lo finaliza:** personal del departamento de formación de la empresa. **Descripción:** a) Si es el departamento de formación quien desea registrar potenciales alumnos de las acciones formativas, se dirige a la zona de nuevos alumnos/as. a.1. En primer lugar, ha de introducir sus datos personales: nombre, apellidos, nif (si tuviera), fecha de nacimiento y sexo. a.2. A continuación, sus datos de contacto: campos de dirección, teléfono/s y email. a.3. Por último, datos generales: categoría (rol que juega en la empresa) y/o observaciones. a.4. El sistema comprobará que todos los datos sean correctos (fecha de nacimiento sea una fecha válida, se han introducido todos los datos obligatorios). Si todo es correcto, el alumno/a queda dado/a de alta en el sistema, y se muestra un mensaje de éxito. Si algún dato es erróneo, se muestra un mensaje indicando los errores. b) [Manual] Si es personal de la empresa quien desea preinscribirse por primera vez en una acción formativa, y no está registrado en la aplicación, lo comunica al departamento de formación. b.1. El departamento de formación se dirige a la edición en la que se desea preinscribir (ver caso de uso 2.2.3.2.2) y, dentro de esta, a la zona de preinscripciones. b.2. Al ver que no se encuentra registrado/a en el sistema, pulsa sobre la opción de nuevo alumno, e introduce sus datos siguiendo los pasos a.1, a.2, a.3 y a.4.

b.3. El sistema da de alta el alumno/a, quedando automáticamente preinscrito/a en la edición.

c) [Manual] Si es personal de la empresa quien desea matricularse por primera vez en una acción formativa, y no está registrado en la aplicación, lo comunica al departamento de formación.
   c.1. El departamento de formación se dirige a la edición en la que se desea matricular (ver caso de uso 2.2.3.2.2) y, dentro de esta, a la zona de matriculaciones.
   c.2. Al ver que no se encuentra registrado/a en el sistema, pulsa sobre la opción de nuevo alumno/a, e introduce sus datos siguiendo los pasos 1.1, 1.2, 1.3 y 1.4.
   c.3. El sistema da de alta el alumno/a, quedando automáticamente matriculado/a en la edición.

Caso de uso: "Buscador de alumnos"

Precondiciones: -
Pos condiciones: listado de alumnos/as.
Quien lo comienza: personal del departamento de formación de la empresa.
Quien lo finaliza: personal del departamento de formación de la empresa.
Descripción:

1. Dirigiéndose a la zona de alumnos/as, se busca uno o varios alumnos/as, pudiendo filtrar por nombre, apellidos, provincia y/o categoría (rol que juega en la empresa).
2. El listado resultante de la búsqueda, podrá ser exportado a un archivo en formato Excel y/o pdf.

Caso de uso: "Eliminación de alumno"

Precondiciones: el alumno/a a eliminar debe estar dado de alta en el sistema.
Pos condiciones: el alumno/a es eliminado del sistema.
Quien lo comienza: personal del departamento de formación de la empresa o el alumno/a.
Quien lo finaliza: personal del departamento de formación de la empresa.
Excepciones: si el alumno/a ya se ha matriculado en alguna edición.
Descripción:

1. [Manual] O bien el alumno/a se pone en contacto con el personal del departamento de formación, o es el propio personal quien toma la decisión de eliminar un/a alumno/a del sistema.
2. Siguiendo el caso de uso 2.2.7.2.2, el personal de formación encuentra el alumno/a que desea borrar.
3. Una vez localizado el alumno/a que se desea borrar, indica al sistema su intención de eliminarlo definitivamente.
4. El sistema nos mostrará el siguiente mensaje de confirmación del borrado: "Los datos borrados no se pueden recuperar. ¿Proceder con el borrado?".
   a) Si se pulsa aceptar, el sistema elimina definitivamente el/la alumno/a y vuelve a mostrar el listado anterior resultante de la búsqueda con los datos actuales (sin el/la alumno/a borrado/a).
   b) Si se pulsa cancelar, el sistema no realiza ninguna acción.

Caso de uso: "Actualización de alumno"
Precondiciones: el alumno/a a actualizar debe estar dado de alta en el sistema. **Pos condiciones:** los datos del alumno/a son actualizados. **Quien lo comienza:** el alumno/a. **Quien lo finaliza:** personal del departamento de formación de la empresa. **Descripción:** 1. [Manual] Un/a alumno/a se pone en contacto con el personal del departamento de formación, comunicando un cambio en sus datos. 2. Mediante el buscador de alumnos/as, se busca el alumno/a que se desea actualizar (ver caso de uso 2.2.7.2.2). 3. Una vez localizado el alumno/a a actualizar, ha de pulsarse sobre el nombre del mismo. 4. El sistema nos mostrará en una pantalla idéntica a la de alta de la alumnos/as, los datos del mismo, pudiendo actualizar los datos necesarios. 5. El sistema comprueba que los datos sean correctos (de la misma manera que alta de alumno/a). a) Si todo es correcto, el sistema guarda los cambios en el/la alumno/a y muestra el mensaje de éxito. b) Si hay algún dato no válido, el sistema deberá de indicar los errores o incoherencias al usuario/a.

3 Modelo de datos del sistema

3.1 Diagrama de estructura de datos

A continuación mostraremos el diagrama de estructuras de datos que hemos realizado para que el sistema pueda trabajar con los datos de la mejor manera posible.

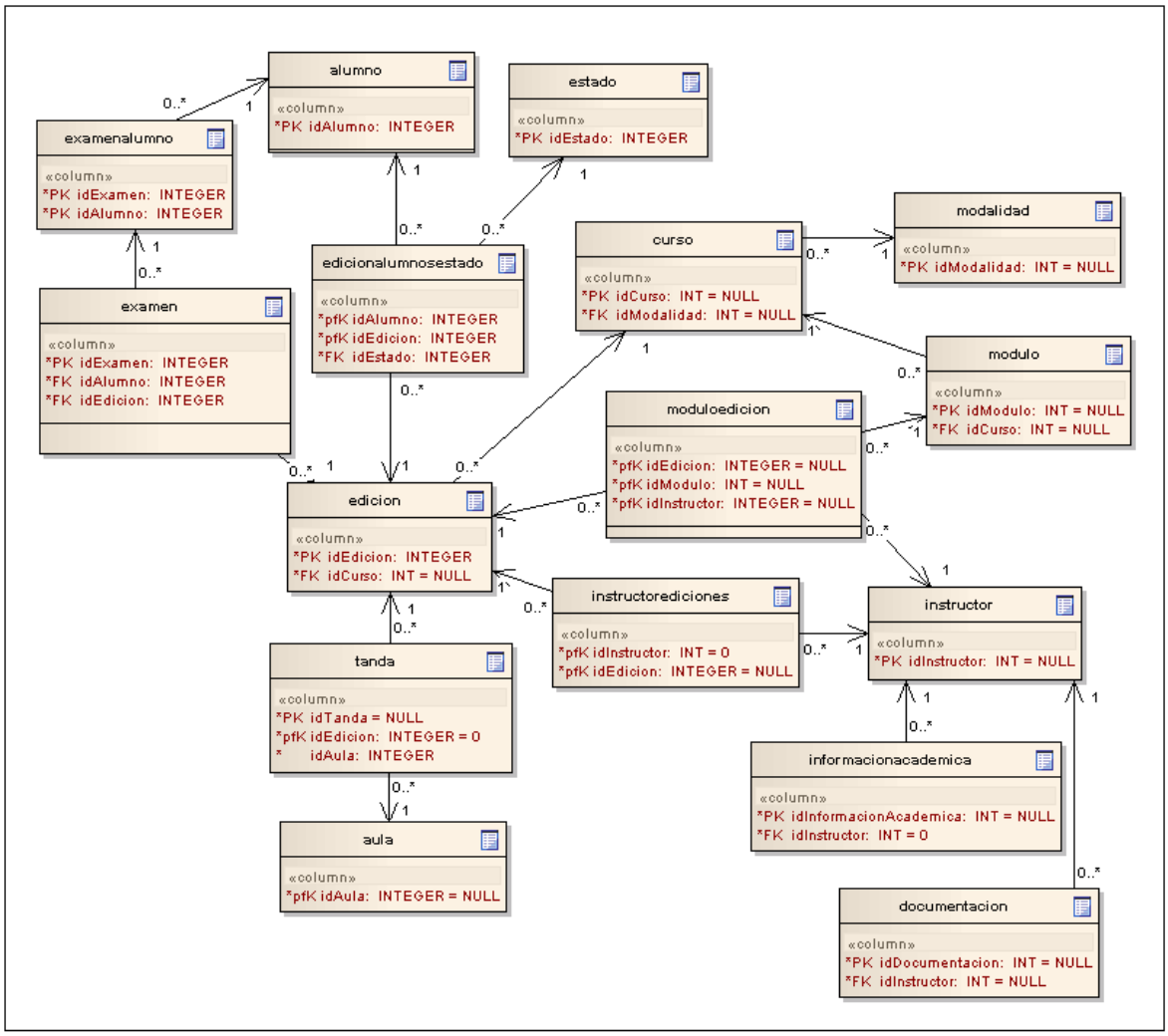


Ilustración 10 - Diagrama de estructura de datos (Enterprise Architect)

4 Interfaces de usuario

4.1 Descripción y mapa de pantallas

Seguidamente se muestran las diferentes pantallas que usará el sistema, a través de las cuales el usuario/a final interactúa en el sistema.

4.1.1 Interfaz inicial

Cuando el usuario/a inicie la aplicación, aparecerá la pantalla inicial, desde donde deberá identificarse para poder acceder al sistema:

Una vez que el usuario/a ha entrado, tendrá acceso a todas las funciones a través del menú superior, que se repetirá a lo largo de todas las pantallas.

4.1.2 Interfaz de cursos

Una vez que el usuario/a accede a la zona de cursos, tendrá acceso a los siguientes interfaces, desde donde podrá gestionar los cursos que se imparten en la empresa.

4.1.2.1 Nuevo curso

Esta es la interfaz para introducir un nuevo curso en el sistema. El usuario/a rellenará los campos (al menos los obligatorios) y podrá guardarlo.

4.1.2.2 Búsqueda de curso

Esta es la interfaz para buscar un curso guardado en el sistema. El usuario/a puede buscar uno o más cursos por su nombre, contenido, modalidad, tipo de curso, provincia, y rangos de horas teóricas y prácticas.

4.1.2.3 Consulta/Modificación de un curso

Se trata de la interfaz de consulta y/o modificación de los datos de un curso guardado en el sistema. Después de realizar la búsqueda, el usuario/a podrá seleccionar un curso concreto.

Posteriormente, aparecerán sus datos.

4.1.2.4 Listado de módulos de los que consta un curso

Este interfaz muestra el listado de módulos de los que consta el curso.

4.1.2.5 Consulta/modificación de un módulo de un curso

Interfaz que muestra los datos del módulo. Puede consultarse para su modificación o simplemente, para una consulta de sus datos.

4.1.2.6 Asignar un nuevo módulo a un curso

El interfaz para crear un módulo del curso, es exactamente igual que el de modificación del módulo (véase 4.3.2.2.6).

4.1.3 Interfaz de ediciones

Una vez que el usuario/a accede a la zona de ediciones, tendrá acceso a los siguientes interfaces, desde donde podrá gestionar las ediciones que se imparten en la empresa.

4.1.3.1 Nueva edición

Esta es la interfaz para introducir una nueva edición en el sistema. El usuario/a rellenará los campos (al menos los obligatorios).

4.1.3.2 Búsqueda de ediciones

Esta es la interfaz para buscar una edición guardada en el sistema. El usuario/a puede buscar una o más ediciones por el curso al que pertenece y rangos de fechas, números de horas teóricas y/o prácticas.

4.1.3.3 Búsqueda de ediciones por provincia

Interfaz para obtener un listado en formato pdf, de las ediciones que aún no han comenzado a impartirse en una provincia.

4.1.3.4 Consulta/Modificación de una edición

Se trata de la interfaz de consulta y/o modificación de los datos de una edición guardada en el sistema. Después de realizar la búsqueda, el usuario/a podrá seleccionar una edición concreta.

Posteriormente, aparecerán sus datos.

4.1.3.5 Preinscripción en una edición

Se trata de la interfaz para preinscribir alumnos/as en una edición, y/o consultar los/as alumnos/as ya preinscritos en la misma.

En la parte superior, se mostrarán los alumnos/as ya preinscritos en la edición, que podrán exportarse como listado en formato Excel o pdf.

En la parte central, aparecerá un buscador de alumnos/as aún no preinscritos/as en la aplicación, pudiendo filtrar por nombre, apellidos, provincia y/o categoría.

Después de realizar la búsqueda de alumnos/as no preinscritos/as, el usuario/a podrá marcar uno o varios alumnos/as para preinscribir.

4.1.3.6 Matriculación de alumnos/as preinscritos/as en una edición

Se trata de la interfaz para matricular alumnos/as preinscritos/as en una edición, y/o consultar los/as alumnos/as ya matriculados en la misma, así como su estado (matriculado / de baja).

En la parte superior de la pantalla, se mostrará un listado de alumnos que ya han sido matriculados/as en la edición. Para mayor usabilidad, se mostrarán en color verde sombreado los que están matriculados, y en color rojo sombreado los que, estando matriculados, han decidido darse de baja en la edición.

En la parte inferior de la pantalla, se han de mostrar los alumnos/as preinscritos y que aún no han sido matriculados en la edición. El usuario/a marcará uno o varios de los alumnos/as, y podrá matricularlos en la edición.

4.1.3.7 Matriculación de alumnos/as no preinscritos/as en una edición

Se trata de la interfaz para matricular alumnos/as que no han sido preinscritos/as en la edición, y/o consultar los/as alumnos/as ya matriculados en la misma, así como su estado (matriculado / de baja).

En la parte superior de la pantalla, se mostrará un listado de alumnos que ya han sido matriculados/as en la edición. Para mayor usabilidad, se mostrarán en color verde sombreado los que están matriculados, y en color rojo sombreado los que, estando matriculados, han decidido darse de baja en la edición.

En la parte central, aparecerá un buscador de alumnos/as aún no matriculados/as en la aplicación, pudiendo filtrar por nombre, apellidos, provincia y/o categoría.

Después de realizar la búsqueda de alumnos/as no matriculados/as, el usuario/a podrá marcar uno o varios alumnos/as para matricular en la edición.

4.1.3.8 Cambio de estado de alumnos/as matriculados en una edición

Interfaz para cambiar el estado (matriculado / de baja) a alumnos/as matriculados/as en una edición.

Se mostrarán todos los alumnos/as matriculados en la misma. Para mayor usabilidad, se mostrarán en color verde sombreado los que están matriculados, y en color rojo sombreado los que, estando matriculados, han decidido darse de baja en la edición.

4.1.3.9 Búsqueda de exámenes en una edición

Esta es la interfaz para buscar exámenes de una edición guardada en el sistema. El usuario/a puede buscar uno o más exámenes por el alumno/a que lo realiza, el tipo de examen, calificación y rangos de puntuaciones y fechas de realización.

Una vez buscados los exámenes de la edición por el/los criterio/s deseado/s, se mostrará un listado de los mismos. Para mayor usabilidad, se mostrarán en color verde sombreado los que ya han sido calificados y obtuvieron una calificación de apto, o una nota superior o igual a 5.0. En color rojo sombreado, se mostrarán aquellos exámenes que hay sido calificados, pero la calificación obtenida no ha llegado a ser apta para superar la prueba.

Los exámenes que aún no se han calificado, no se verán sombreados.

4.1.3.10 Creación de exámenes en una edición

Esta es la interfaz para crear exámenes en una edición. Para ello, ha de seleccionarse el tipo de examen, la fecha de realización, y uno o varios alumnos/as que realizarán dicho examen.

4.1.3.11 Calificación de exámenes

Interfaz para calificar los exámenes ya realizados, en una edición. Para ello, el usuario/a, podrá filtrar los exámenes por alumno/a que lo ha realizado, el tipo de examen y un rango de fechas de realización del mismo.

Una vez filtrada la búsqueda, podrá introducir la calificación y/o observaciones de aquellos exámenes que desee calificar.

Se mostrarán en color verde sombreado los que ya han sido calificados y obtuvieron una calificación de apto, o una nota superior o igual a 5.0. En color rojo sombreado, se mostrarán aquellos exámenes que hay sido calificados, pero la calificación obtenida no ha llegado a ser apta para superar la prueba.

Los exámenes que aún no se han calificado, no se verán sombreados.

4.1.3.12 Creación/modificación del horario de impartición de una edición

Interfaz para crear o modificar el horario de impartición de una edición. Para ello, el usuario/a, podrá seleccionar en el calendario un día concreto, o un rango de fechas. Posteriormente, filtrar ese rango de fechas por días de la semana y, finalmente, introducir la hora de inicio y fin, el aula en el que se impartirá y si se trata de parte teórica o práctica.

4.1.3.13 Listado de módulos de los que consta una edición

Esta interfaz muestra los módulos que forman parte de una edición.

4.1.3.14 Actualización de datos de los módulos de una edición

Interfaz para asignar o actualizar el instructor/a asociado a cada módulo de la edición, así como las fechas de inicio y fin de impartición de cada módulo.

4.1.3.15 Asignar instructores/as a una edición

Este interfaz, muestra en primer lugar los instructores/as ya asignados a la edición. Si se desea asignar algún otro/a instructor, se puede filtrar los instructores/as aún no asignados por nombre, apellidos, tipo de instructor/a, y rangos de precio por hora.

A continuación, se puede marcar uno o varios instructores/as para asociar a la edición.

4.1.3.16 Actualizar datos de instructores/as asignados a una edición

Este interfaz, para actualizar los datos de los instructores/as ya asignados a la edición. Podrá introducir el precio/hora y/o el tipo de instructor (coordinador/profesor) para esa formación, en uno o varios instructores/as.

4.1.4 Interfaz de alumnos/as

Una vez que el usuario/a accede a la zona de alumnos/as, tendrá acceso a los siguientes interfaces, desde donde podrá gestionar los/as alumnos/as que recibirán las formaciones o que son candidatos a recibirlas.

4.1.4.1 Alta de alumno/a

Esta es la interfaz para introducir un nuevo alumno/a en el sistema.

Los datos del alumno/a estarán divididos en tres secciones: datos personales, datos de contacto, y datos generales.

4.1.4.2 Búsqueda de alumno/a

Esta es la interfaz para buscar alumnos/as registrados/as en el sistema. El usuario/a puede buscar uno o más alumnos/as por su nombre, apellidos, provincia y/o categoría.

4.1.4.3 Consulta/Modificación de un alumno/a

Se trata de la interfaz de consulta y/o modificación de los datos de un alumno/a registrado/a en el sistema. Después de realizar la búsqueda, el usuario/a podrá seleccionar un alumno concreto.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de alumnos/as, pero con los datos rellenos.

4.1.5 Interfaz de aulas

Una vez que el usuario/a accede a la zona de aulas, tendrá acceso a los siguientes interfaces, desde donde podrá gestionar las aulas en las que se impartirán las actividades formativas.

4.1.5.1 Nueva aula

Esta es la interfaz para introducir un aula nueva en el sistema. El usuario/a rellenará los campos (al menos los obligatorios) y podrá guardarla.

Los datos del aula estarán divididos en tres secciones: datos generales, datos de contacto, y valoración / observaciones.

4.1.5.2 Búsqueda de aula

Esta es la interfaz para buscar aulas registradas en el sistema. El usuario/a puede buscar uno o más aulas por su nombre, provincia, municipio y población donde se encuentra y rangos de número de puestos, número de metros y valoración media.

4.1.5.3 Consulta/Modificación de un aula

Se trata de la interfaz de consulta y/o modificación de los datos de un aula registrado/a en el sistema. Después de realizar la búsqueda, el usuario/a podrá seleccionar un aula concreta.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de aulas, pero con los datos rellenos.

4.1.6 Interfaz de instructores/as

Una vez que el usuario/a accede a la zona de instructores/as, tendrá acceso a los siguientes interfaces, desde donde podrá gestionar los/as instructores/as que impartirán las formaciones o que son candidatos a impartirlas.

4.1.6.1 Alta de instructor/a

Esta es la interfaz para introducir un nuevo instructor/a en el sistema. Los datos del instructor/a estarán divididos en tres secciones: datos personales, datos de contacto, y datos generales.

4.1.6.2 Búsqueda de instructor/a

Esta es la interfaz para buscar instructores/as registrados/as en el sistema. El usuario/a puede buscar uno o más instructores/as por su nombre, apellidos, provincia, municipio, población, tipo de instructor (laboral o interno) y rango de valoraciones medias y de precio por hora.

4.1.6.3 Consulta/Modificación de un instructor/a

Se trata de la interfaz de consulta y/o modificación de los datos de un instructor/a registrado/a en el sistema. Después de realizar la búsqueda, el usuario/a podrá seleccionar un instructor concreto.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de instructores/as, pero con los datos rellenos.

4.1.6.4 Listado de información académica de un instructor/a

Este interfaz muestra el listado de información académica que posee un instructor/a.

4.1.6.5 Consulta/modificación de una formación académica de un instructor/a

Interfaz que muestra los datos de una formación académica de un instructor/a concreto/a. Puede consultarse para su modificación o simplemente, para una consulta de sus datos.

4.1.6.6 Recoger información académica de un instructor/a

En este interfaz, se puede registrar una formación académica para el instructor/a. Es idéntica al interfaz de consulta/modificación de instructor/a.

4.1.6.7 Listado de documentación de un instructor/a

Este interfaz muestra el listado de información académica que posee un instructor/a.

4.1.6.8 Consulta/modificación de documentación de un instructor/a

Interfaz que muestra los datos de una documentación de un instructor/a concreto/a. Puede consultarse para su modificación o simplemente, para una consulta de sus datos.

4.1.6.9 Recoger documentación de un instructor/a

En este interfaz, se puede registrar la documentación disponible del instructor/a.

4.1.6.10 Interfaz de administración

Una vez que el usuario/a accede a la zona de administración, tendrá acceso a los siguientes interfaces, desde donde podrá tratar las gestiones básicas, necesarias para el resto de interfaces. También podrá administrar la gestión de usuarios/as de la aplicación.

4.1.6.11 Listado de categorías

Esta es la interfaz donde se ven las categorías que se han introducido en el sistema.

4.1.6.12 Alta de categoría

Esta es la interfaz para introducir una nueva categoría en el sistema. El usuario/a rellenará la descripción de la categoría y podrá guardarla en el sistema.

4.1.6.13 Consulta/Modificación de una categoría

Se trata de la interfaz de consulta y/o modificación de los datos de una categoría registrada en el sistema. El usuario/a podrá seleccionar una categoría concreta en el listado de las mismas.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de categorías, pero con los datos rellenos.

4.1.6.14 Listado de homologaciones

Esta es la interfaz donde se ven las homologaciones que se han introducido en el sistema. Tiene exactamente el mismo formato que el interfaz del listado de categorías (3.3.2.7.1), pero añadiendo dos campos más: tipo de homologación y observaciones.

4.1.6.15 Alta de homologaciones

Esta es la interfaz para introducir una nueva homologación en el sistema, que posteriormente, se podrá asignar a un aula. El usuario/a deberá rellenar, al menos, la descripción de la homologación y el tipo (el campo observaciones no es obligatorio) y podrá guardarla en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1), pero añadiendo dos campos más: tipo de homologación, con formato de lista desplegable, y observaciones, con formato de edición de texto.

4.1.6.16 Consulta/Modificación de una homologación

Se trata de la interfaz de consulta y/o modificación de los datos de una homologación registrada en el sistema. El usuario/a podrá seleccionar una homologación concreta en el listado de las mismas.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de homologaciones, pero con los datos rellenos.

4.1.6.17 Listado de modalidades

Esta es la interfaz donde se ven las modalidades que se han introducido en el sistema. Tiene exactamente el mismo formato que el interfaz del listado de categorías (3.3.2.7.1)

4.1.6.18 Alta de modalidad

Esta es la interfaz para introducir una nueva modalidad en el sistema. El usuario/a rellenará la descripción de la modalidad y podrá guardarla en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

4.1.6.19 Consulta/Modificación de una modalidad

Se trata de la interfaz de consulta y/o modificación de los datos de una modalidad registrada en el sistema. El usuario/a podrá seleccionar una modalidad concreta en el listado de las mismas.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de modalidades, pero con los datos rellenos.

4.1.6.20 Listado de tipos de documento

Esta es la interfaz donde se ven los tipos de documento que se han introducido en el sistema. Tiene exactamente el mismo formato que el interfaz del listado de categorías (3.3.2.7.1)

4.1.6.21 Alta de tipos de documento

Esta es la interfaz para introducir un nuevo tipo de documento en el sistema. El usuario/a rellenará la descripción del tipo de documento y podrá guardarlo en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

4.1.6.22 Consulta/Modificación de un tipo de documento

Se trata de la interfaz de consulta y/o modificación de los datos de un tipo de documento registrado en el sistema. El usuario/a podrá seleccionar un tipo de documento concreto en el listado de los mismos.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de tipos de documento, pero con los datos rellenos.

4.1.6.23 Listado de tipos de examen

Esta es la interfaz donde se ven los tipos de examen que se han introducido en el sistema. Tiene exactamente el mismo formato que el interfaz del listado de categorías (3.3.2.7.1)

4.1.6.24 Alta de tipos de examen

Esta es la interfaz para introducir un nuevo tipo de examen en el sistema. El usuario/a rellenará la descripción del tipo de examen y podrá guardarlo en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

4.1.6.25 Consulta/Modificación de un tipo de examen

Se trata de la interfaz de consulta y/o modificación de los datos de un tipo de examen registrado en el sistema. El usuario/a podrá seleccionar un tipo de documento concreto en el listado de los mismos.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de tipos de examen, pero con los datos rellenos.

4.1.6.26 Listado de tipos de homologación

Esta es la interfaz donde se ven los tipos de homologación que se han introducido en el sistema. Tiene exactamente el mismo formato que el interfaz del listado de categorías (3.3.2.7.1)

4.1.6.27 Alta de tipos de homologación

Esta es la interfaz para introducir un nuevo tipo de homologación en el sistema. El usuario/a rellenará la descripción del tipo de homologación y podrá guardarlo en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

4.1.6.28 Consulta/Modificación de un tipo de homologación

Se trata de la interfaz de consulta y/o modificación de los datos de un tipo de homologación registrado en el sistema. El usuario/a podrá seleccionar un tipo de homologación concreto en el listado de los mismos.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de tipos de homologación, pero con los datos rellenos.

4.1.6.29 Listado de tipos de vía

Esta es la interfaz donde se ven los tipos de vía que se han introducido en el sistema. Tiene exactamente el mismo formato que el interfaz del listado de categorías (3.3.2.7.1)

4.1.6.30 Alta de tipos de vía

Esta es la interfaz para introducir un nuevo tipo de vía en el sistema. El usuario/a rellenará la descripción del tipo de vía y podrá guardarlo en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

4.1.6.31 Consulta/Modificación de un tipo de vía

Se trata de la interfaz de consulta y/o modificación de los datos de un tipo de vía registrado en el sistema. El usuario/a podrá seleccionar un tipo de vía concreto en el listado de los mismos.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de tipos de vías, pero con los datos rellenos.

4.1.6.32 Buscador de usuarios/as

Esta es la interfaz para buscar usuarios/as registrados/as en el sistema. El usuario/a puede buscar uno o más usuarios/as por su nombre, apellidos, login y estado (activo, inactivo).

4.1.6.33 Alta de usuarios/as

Esta es la interfaz para introducir un nuevo usuario en el sistema. El usuario/a rellenará el nombre y apellidos, login, contraseña, confirmación de la contraseña, teléfono, móvil y email, y podrá guardarlo en el sistema. Si se quiere que el nuevo usuario/a pueda funcionar con la aplicación, deberá marcar, además, el check de activo.

4.1.6.34 Consulta/Modificación de un usuario

Se trata de la interfaz de consulta y/o modificación de los datos de un usuario registrado en el sistema. El usuario/a podrá seleccionar un usuario/a concreto en el listado de los mismos, resultante de la búsqueda.

Posteriormente, aparecerán sus datos, en un interfaz idéntico al de alta de usuarios, pero con los datos rellenos.

4.1.6.35 Listado de provincias

Esta es la interfaz donde se ven las provincias que se han introducido en el sistema.

4.1.6.36 Alta de provincias

Esta es la interfaz para introducir una nueva provincia en el sistema. El usuario/a rellenará el nombre de la provincia y podrá guardarla en el sistema.

La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

4.1.6.37 Consulta/Modificación de una provincia, nuevo municipio

Se trata de la interfaz de consulta y/o modificación de los datos de una provincia registrada en el sistema. El usuario/a podrá seleccionar una provincia concreta en el listado de las mismas.

Posteriormente, se verán su descripción, que podrá modificarse si así se desea, y el listado de municipios que la forman. En la parte inferior de la pantalla, se da la opción de añadir un nuevo municipio a la provincia.

4.1.6.38 Consulta/Modificación de un municipio, nueva población

Se trata de la interfaz de consulta y/o modificación de los datos de un municipio registrado en el sistema. El usuario/a podrá seleccionar un municipio concreto en el listado de las mismas.

Posteriormente, se verán su descripción, que podrá modificarse si así se desea, y el listado de poblaciones que la forman. En la parte inferior de la pantalla, se da la opción de añadir una nueva población al municipio.

La apariencia de esta interfaz es similar a la de consulta/modificación de una provincia, nuevo municipio (3.7.2.7.33).

4.1.6.39 Modificar población

Se trata de la interfaz de consulta y/o modificación de los datos de una población registrada en el sistema. El usuario/a podrá seleccionar una población concreta en el listado de las mismas.La apariencia de esta interfaz, es la misma que la de alta de categorías (3.3.2.7.1)

5 Diseño de la arquitectura del sistema

5.1 Especificación de subsistemas de diseño

5.1.1 Diagrama de Paquetes

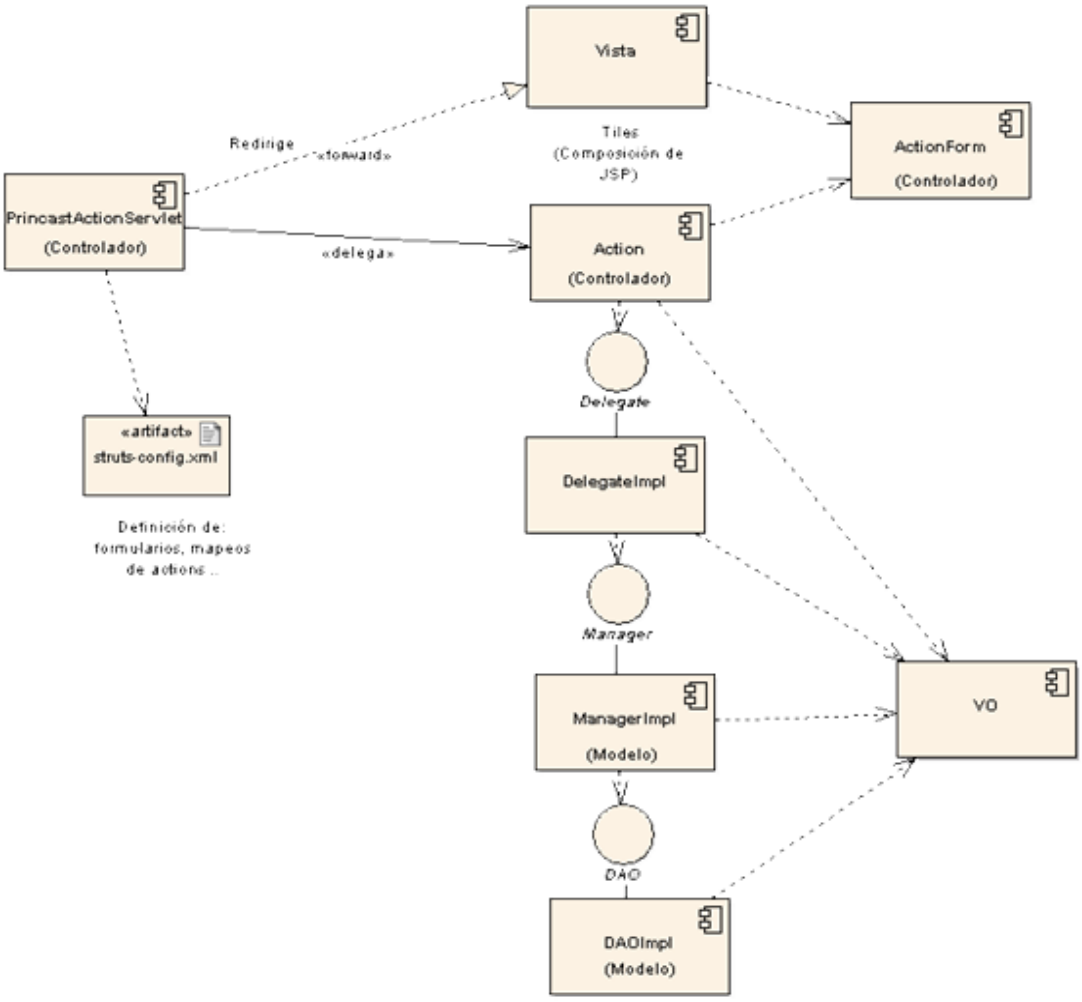


Ilustración 11 - Diagrama de paquetes general

5.1.2 División en subsistemas de diseño

Pasamos a detallar la estructura interna de cada subsistema, definiendo sus interfaces, mostrando las unidades de las que se componen y las clases que pertenecen a cada unidad. Delas clases que componen las unidades, detallaremos profundamente las que tenga una funcionalidad más compleja.

De los interfaces especificaremos los datos que se intercambian, los valores de esos datos y su origen y destino.

Para la comunicación entre capas, utilizaremos objetos VO (Value Object), un patrón de diseño muy utilizado para la transferir información entre las capas de la aplicación, por lo que también se suelen denominar DTO. (Data Transfer Object).

Los VO son una imagen del modelo de datos, aunque no tienen por qué ser exactamente iguales que la estructura de la base de datos. Por ejemplo, a continuación se muestra el Value Object EdicionVO, que como vemos, tiene los campos fechaInicioDesde, fechaInicioHasta, que no aparecen en la tabla ediciones de la base de datos pero que, nos son muy útiles para las búsquedas de una edición. Por tanto los Value Object son una abstracción del modelo de datos de la base de datos.

5.2 Diseño detallado de los subsistemas de diseño

A continuación vamos a ver los diagramas que representan los interfaces de la aplicación. Los interfaces nos muestran los métodos que podemos utilizar de cada módulo, por lo que son la puerta de entrada a la implementación.

5.2.1 Diseño detallado de los subsistemas específicos

5.2.1.1 Subsistema Cursos

Veamos los diagramas principales del subsistema cursos, seguido de diagramas con la definición de los interfaces y la especificación de las clases que los implementan.

5.2.1.1.1 Clases Action

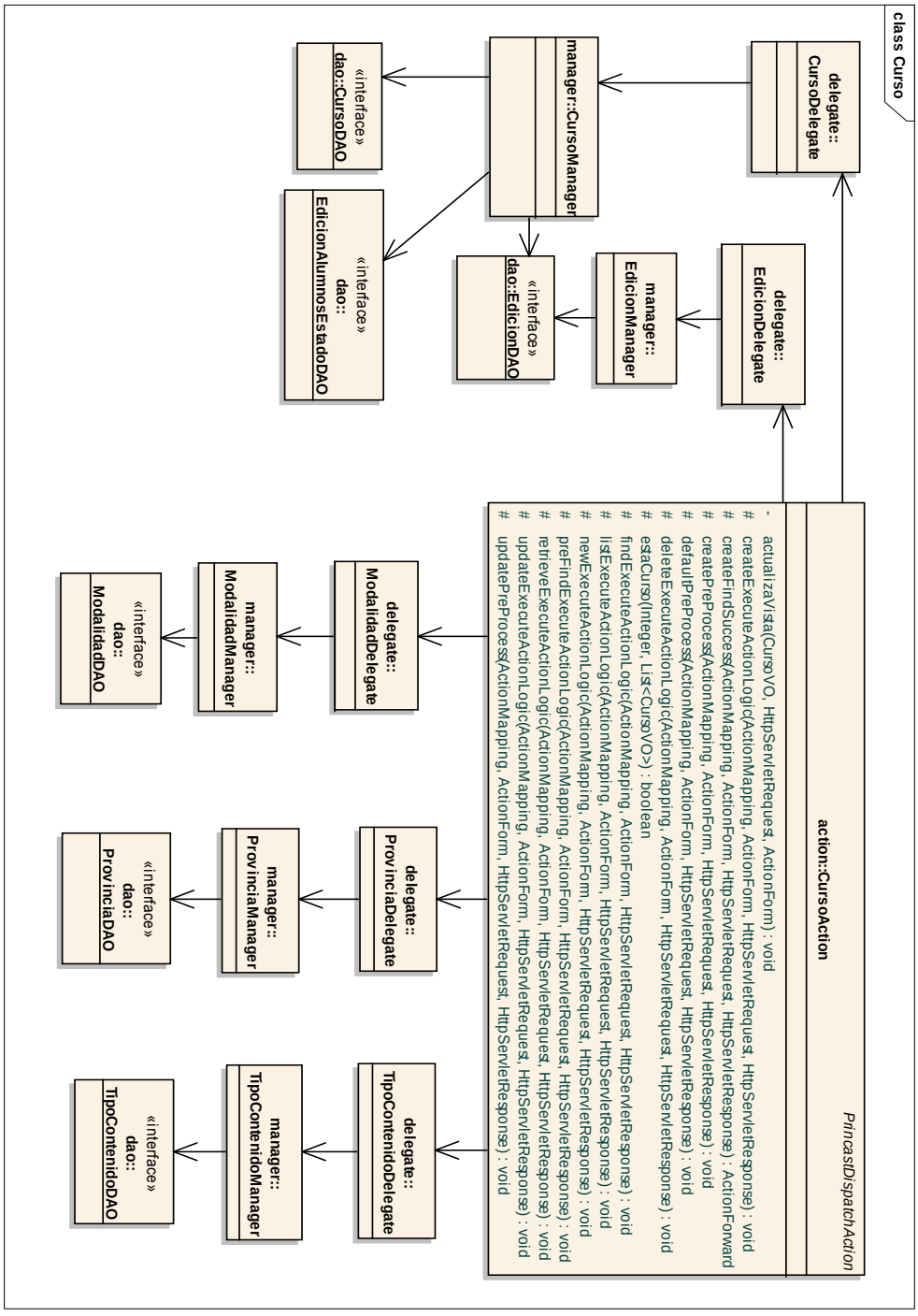


Ilustración 12 - Diagrama de clases principal del subsistema cursos

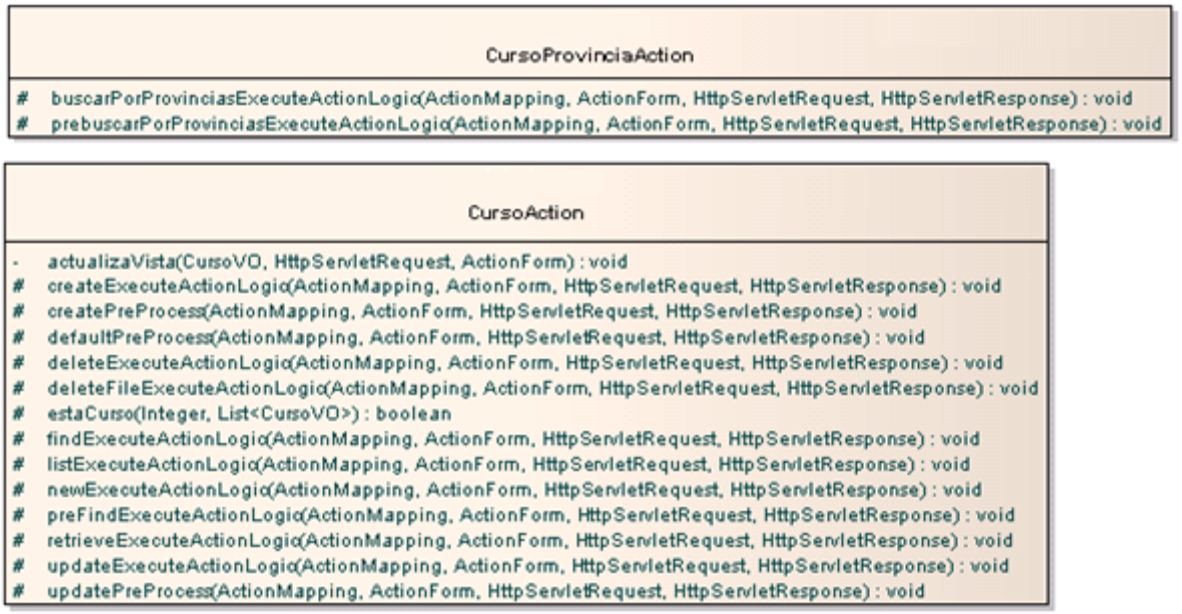


Ilustración 13 - Clases Action del subsistema cursos

5.2.1.1.2 Interfaces Delegate

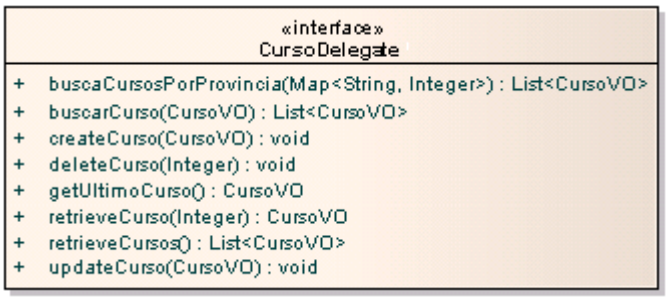


Ilustración 14 - Interfaces Delegate del subsistema cursos

5.2.1.1.3 Interfaces Manager

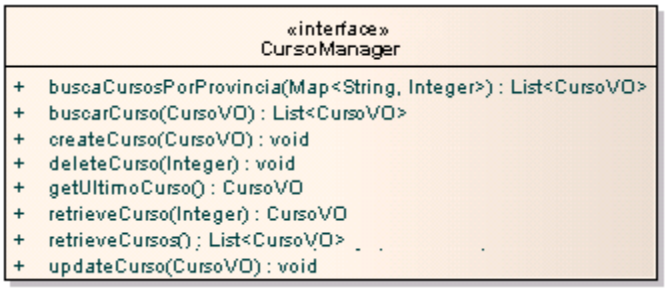


Ilustración 15 - Interfaces Manager del subsistema cursos

5.2.1.1.4 Interface DAO

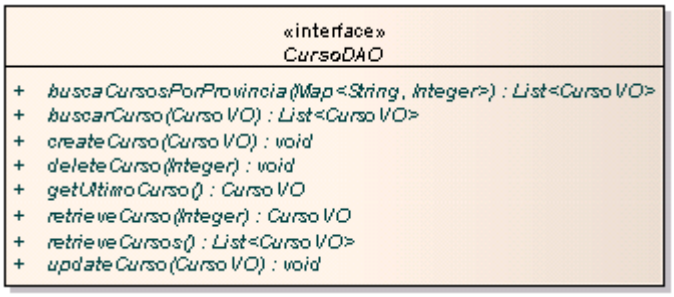


Ilustración 16 - Interfaces DAO del subsistema cursos

5.2.1.1.5 Diagramas de secuencia

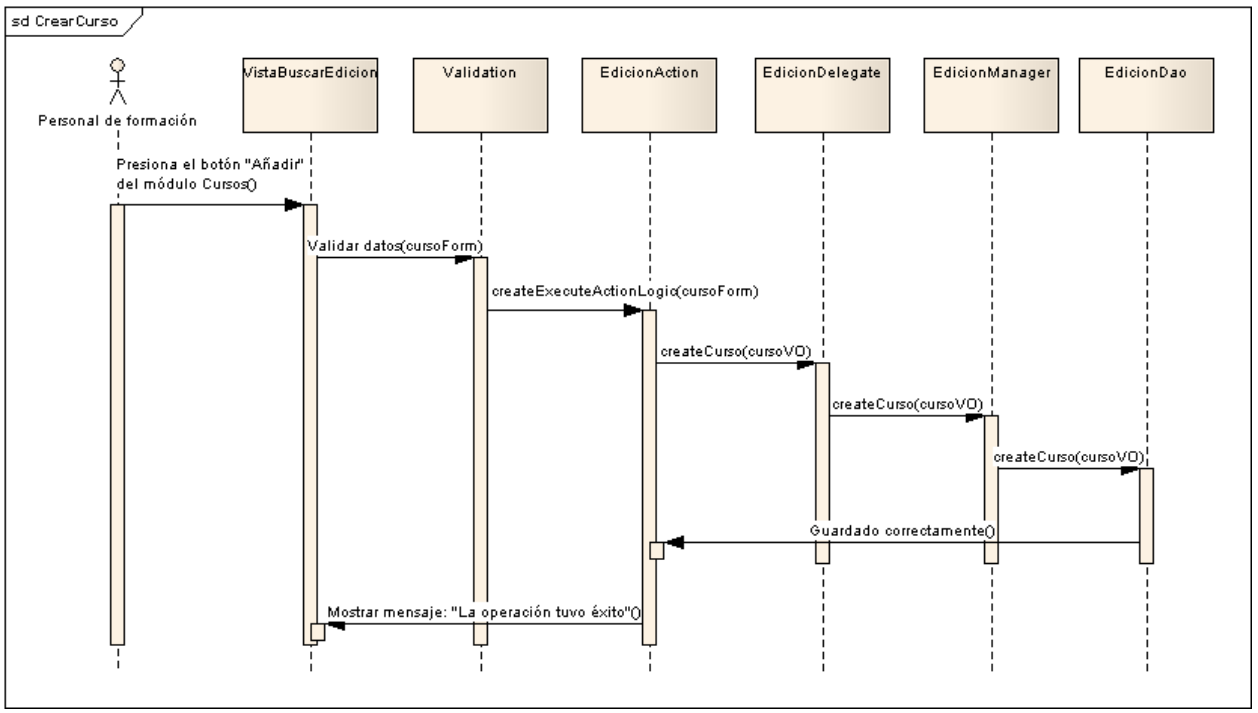


Ilustración 17 - Diagrama de secuencia: guardar curso


EdicionAction
- actualizaVista(EdicionVO, HttpServletRequest, ActionForm) : void
buscarExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
createExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
createPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
defaultPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
deleteExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
listExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
prebuscarExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
retrieveExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
updateExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
updatePreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
PreinscribirAction
inscribirExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preBusquedaExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preInscribirExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
ModuloEdicionAction
createExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
createModulosEdicionExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
createModulosEdicionPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
+ defaultPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
# deleteExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
# listExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
# listModulosExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
# newExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
# retrieveExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
# updateExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
MatricularPreinscritosAction
defaultPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
matricularExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
matricularPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preMatricularExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
MatricularNoPreinscritosAction
defaultPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
matricularExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
matricularPreProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preBusquedaExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preMatricularExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
MatriculadosCambiarEstadoAction
cambiarEstadoExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
despuesCambiarEstadoExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
listarExecuteActionLogic(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void
preProcess(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) : void

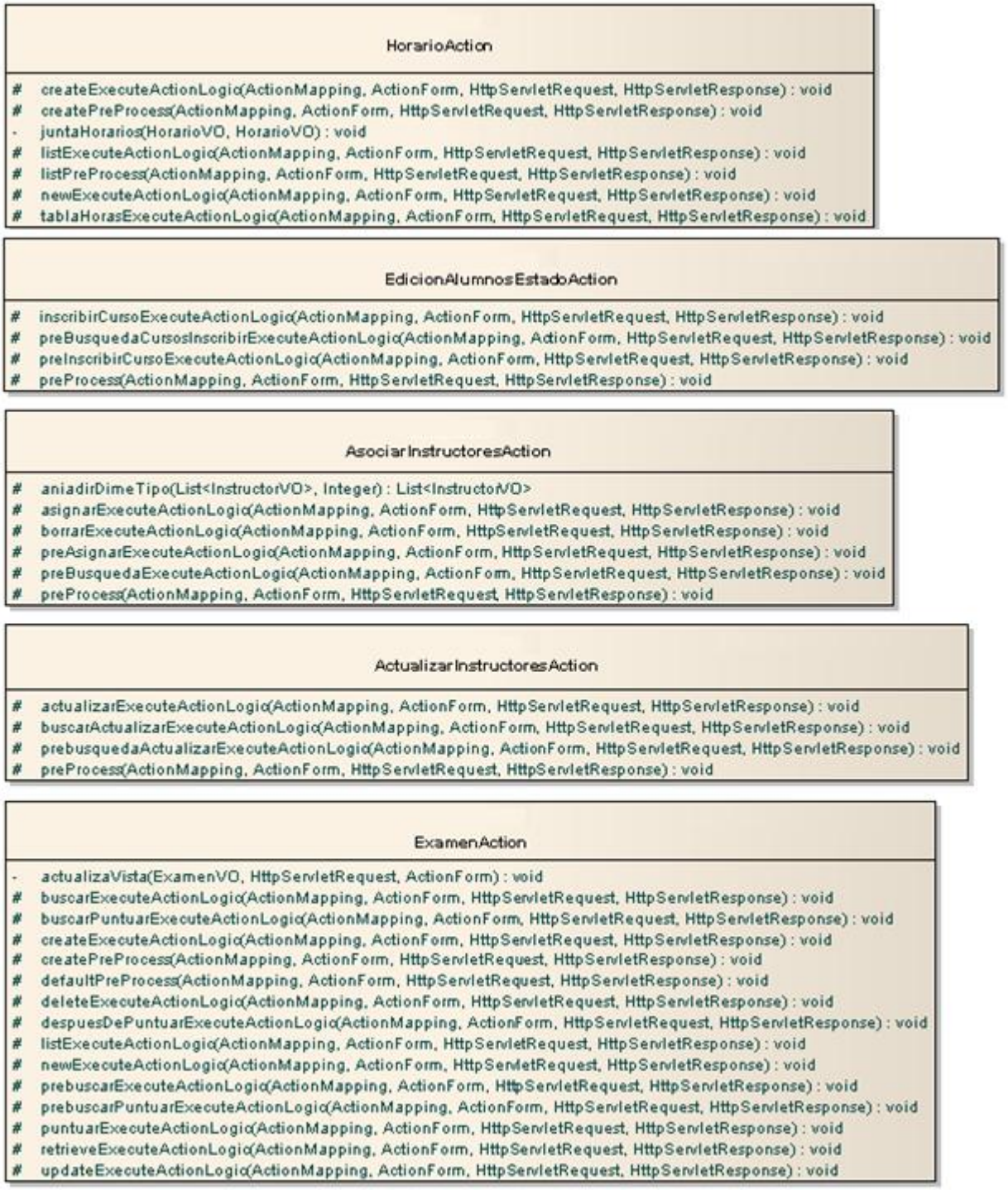


Ilustración 18 - Clases Action del subsistema ediciones

5.2.1.1.6 Interfaces Manager

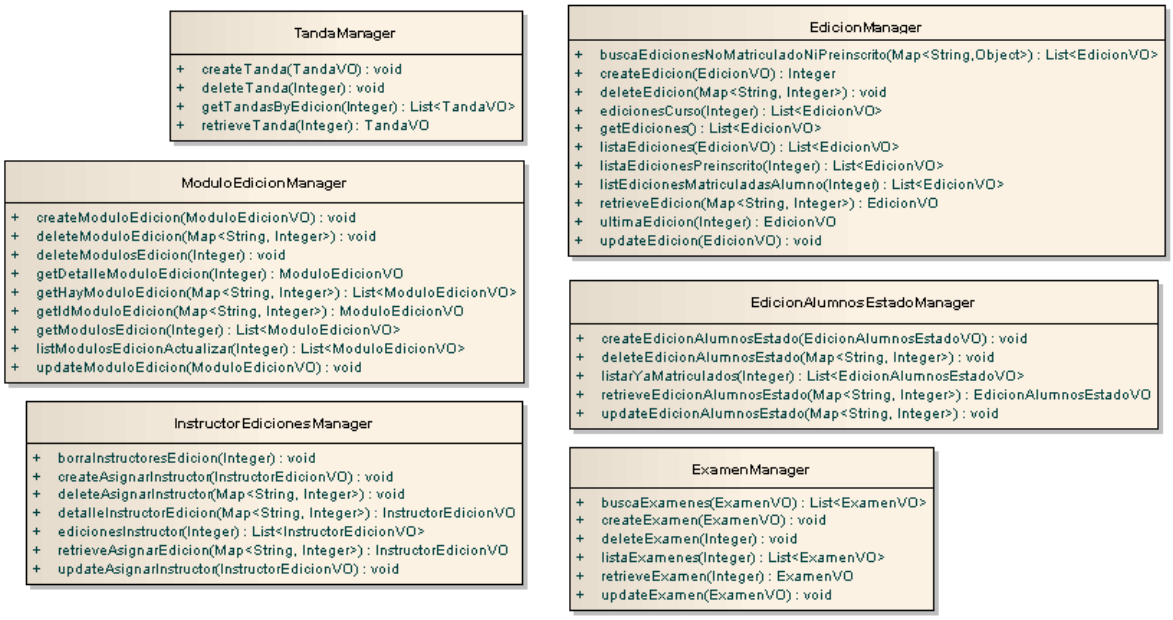


Ilustración 19 - Interfaces Manager del subsistema ediciones

5.2.1.1.7 Interface DAO

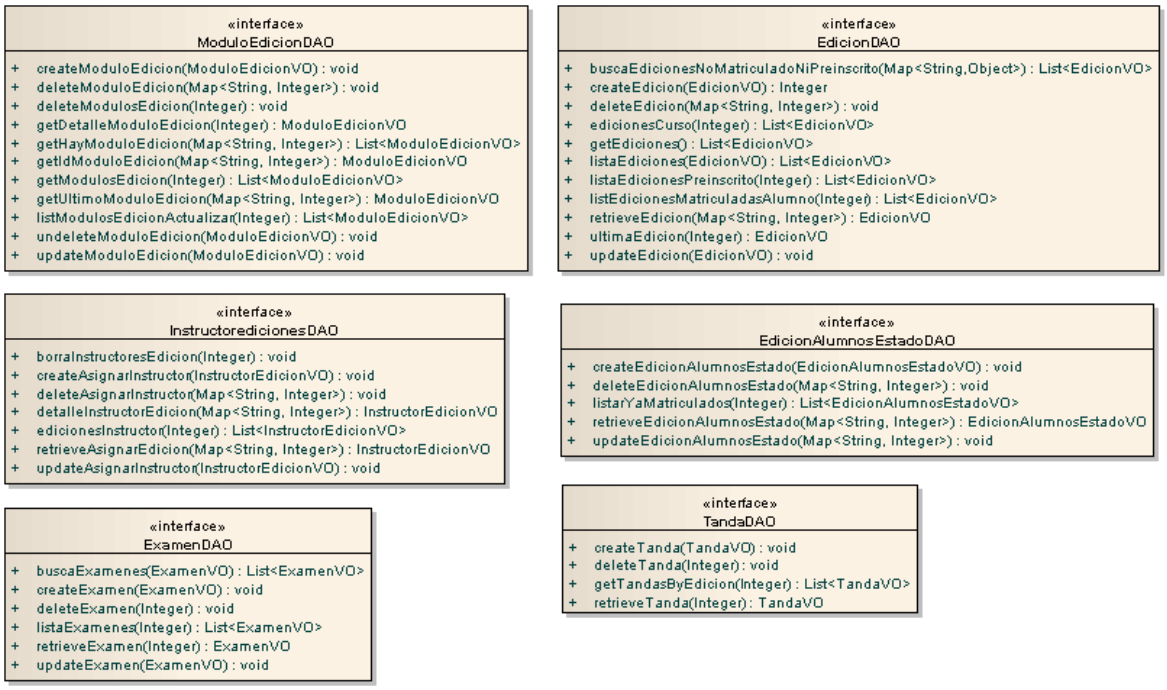


Ilustración 20 - Interfaces DAO del subsistema ediciones

Diagramas de secuencia

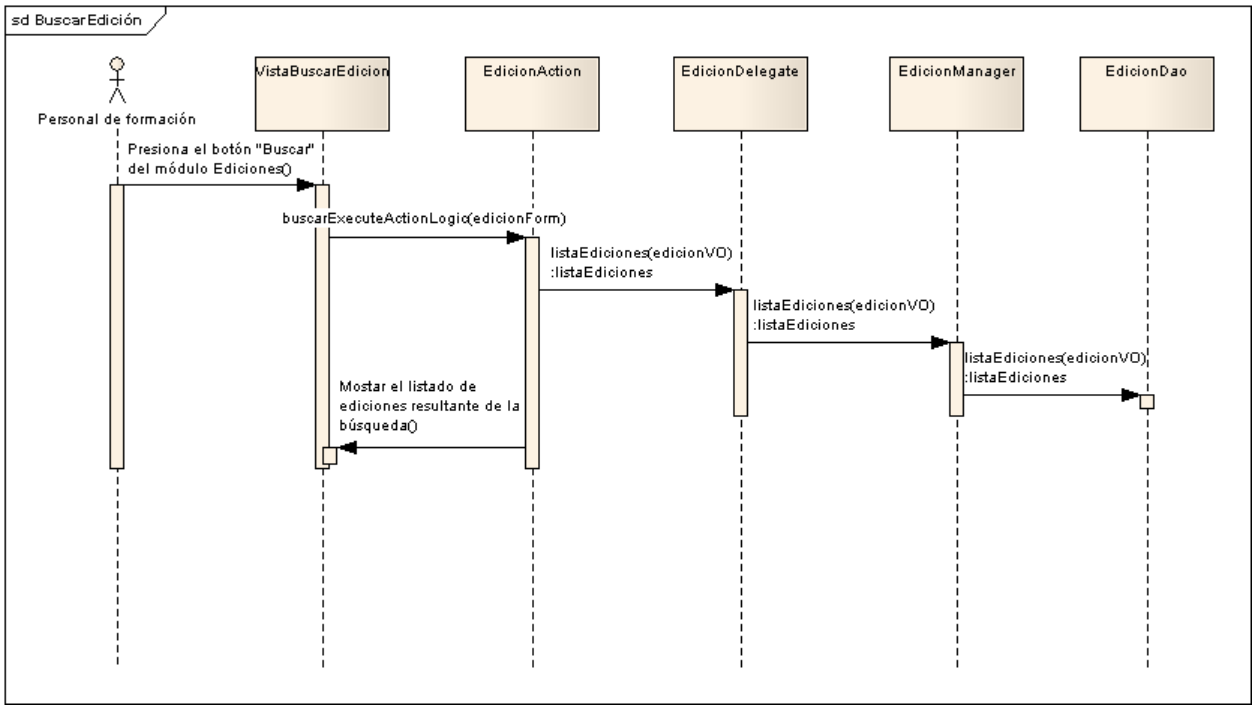


Ilustración 21 - Diagrama de secuencia: buscar ediciones

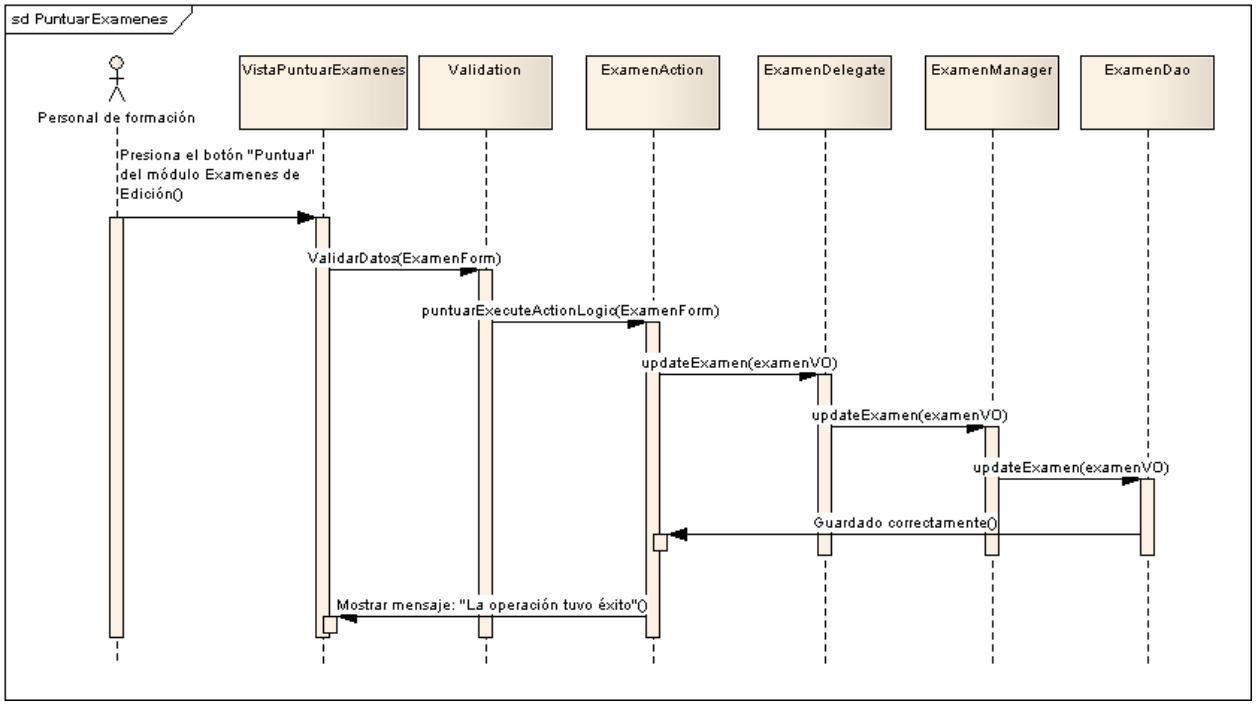


Ilustración 22 - Diagrama de secuencia: puntuar exámenes

A continuación vamos a ver el diagrama de secuencia para preinscribir a alumnos/as en una edición, teniendo en cuenta que para la matriculación, tanto de alumnos/as preinscritos como no preinscritos, es totalmente análogo:

5.2.1.2 Subsistema Alumnos/as

5.2.1.2.1 Clases Action

Ilustración 23 - Clases Action del subsistema alumnos/as

5.2.1.2.2 Inerfaces Manager

Ilustración 24 - Interfaces Manager del subsistema alumnos/as

5.2.1.2.3 Interface DAO

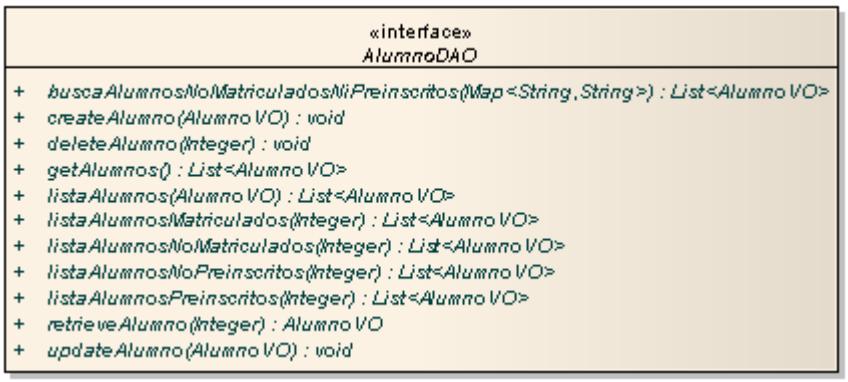


Ilustración 25 - Interfaces DAO del subsistema alumnos/as

5.2.1.2.4 Diagramas de secuencia

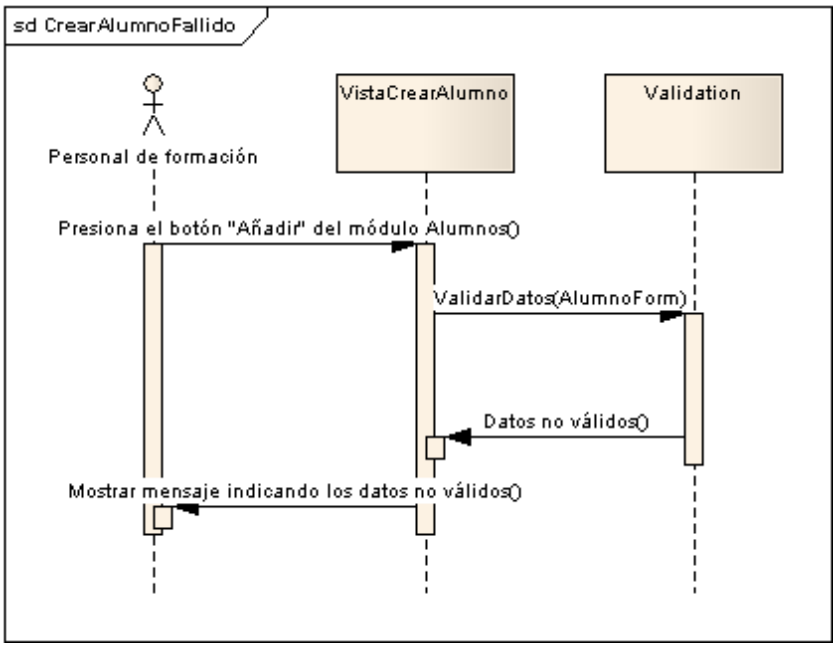


Ilustración 26 - Diagrama de secuencia: crear alumnos/as no válidos

5.2.1.3 Subsistema Aulas

5.2.1.3.1 Clases Action

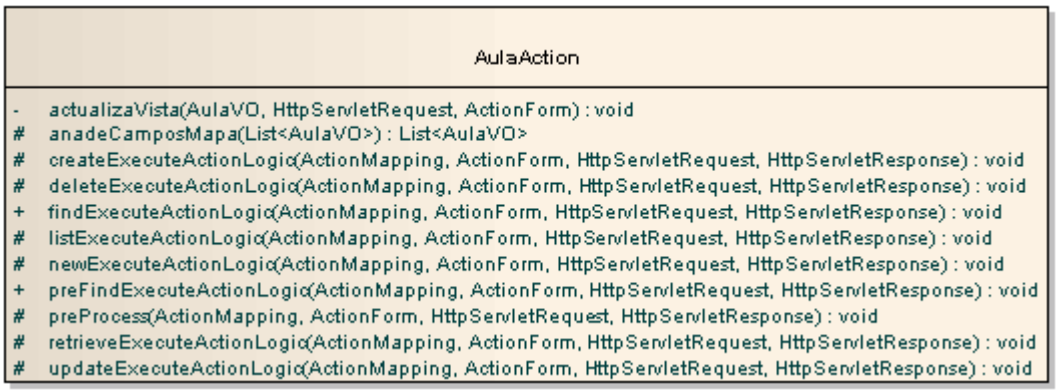


Ilustración 27 - Clases Action del subsistema aulas

5.2.1.3.2 Inerfaces Manager

Ilustración 28 - Interfaces Manager del subsistema aulas

5.2.1.3.3 Interface DAO

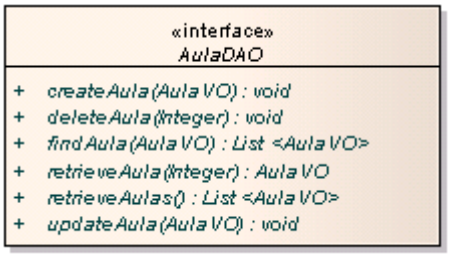


Ilustración 29 - Interfaces DAO del subsistema aulas

5.2.1.3.4 Diagramas de secuencia

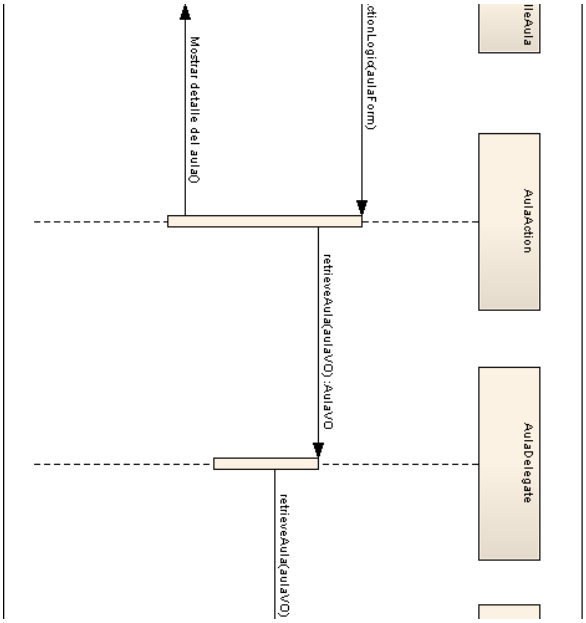


Ilustración 30 - Diagrama de secuencia: ver el detalle de un aula

5.2.1.4 Subsistema Instructores/as

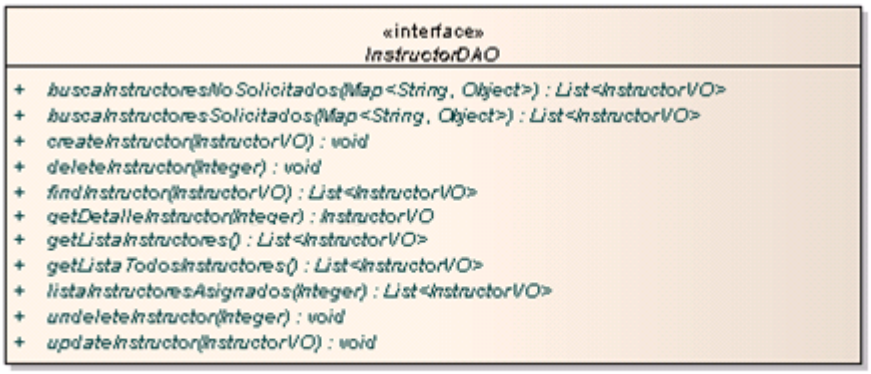
5.2.1.4.1 Clases Action

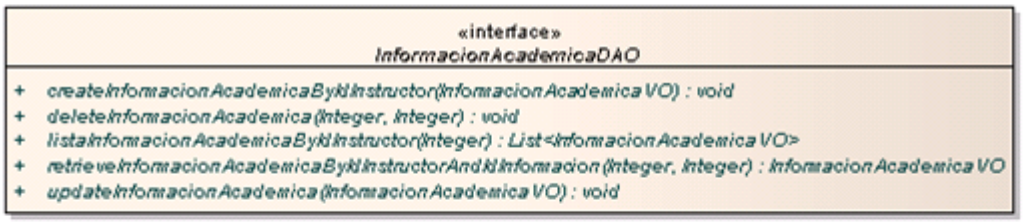
Ilustración 31 - Clases Action del subsistema instructores/as

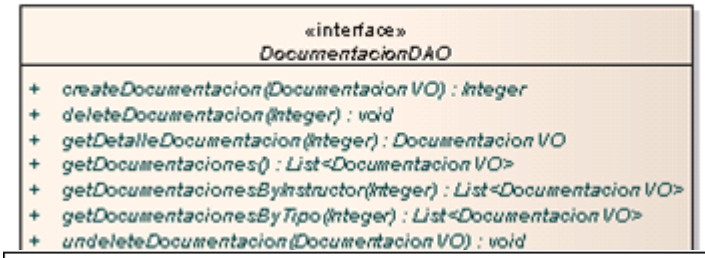
5.2.1.4.2 Interfaces Manager

Ilustración 32 - Interfaces Manager del subsistema instructores/as

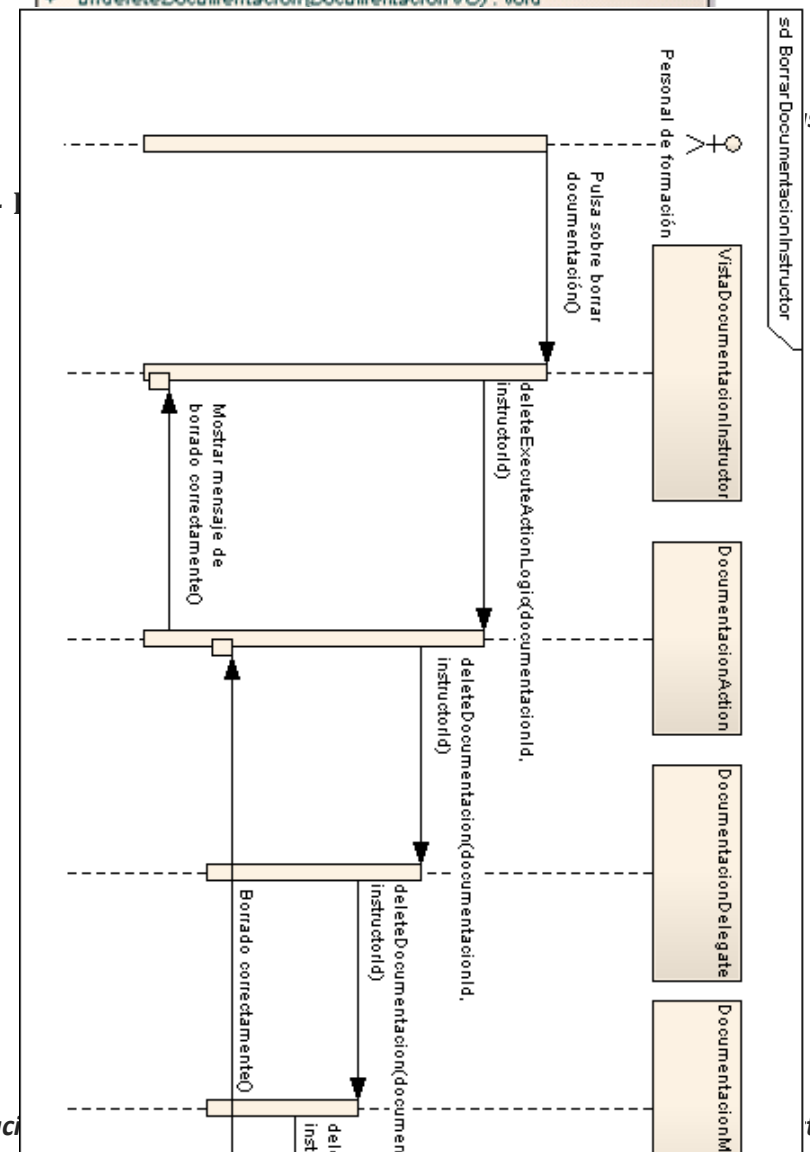
5.2.1.4.3 Interface DAO







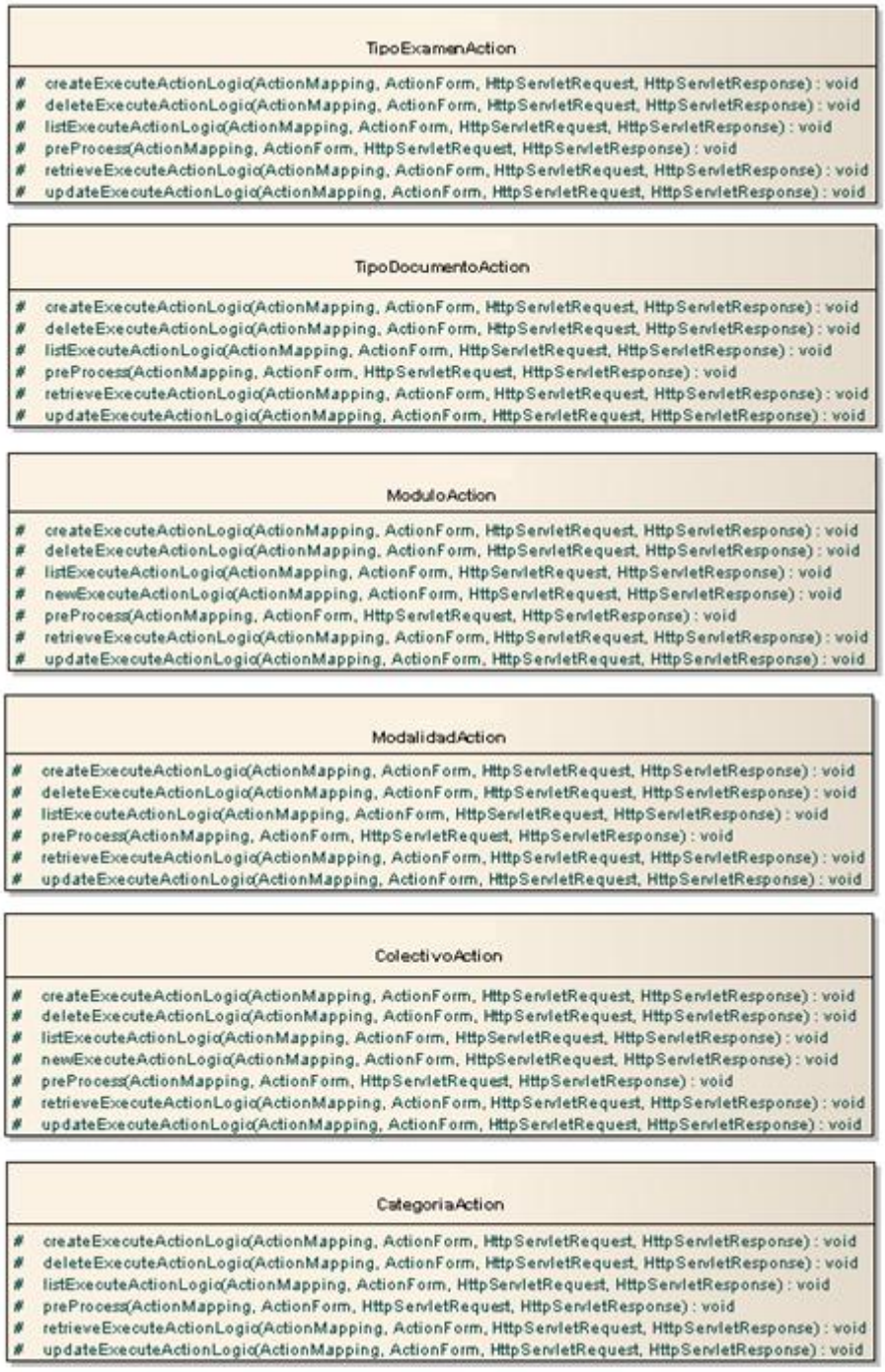
5.2.1.4.4 I



Ilustraci *tor/a*

5.2.1.5 Subsistema Administración

5.2.1.5.1 Clases Action



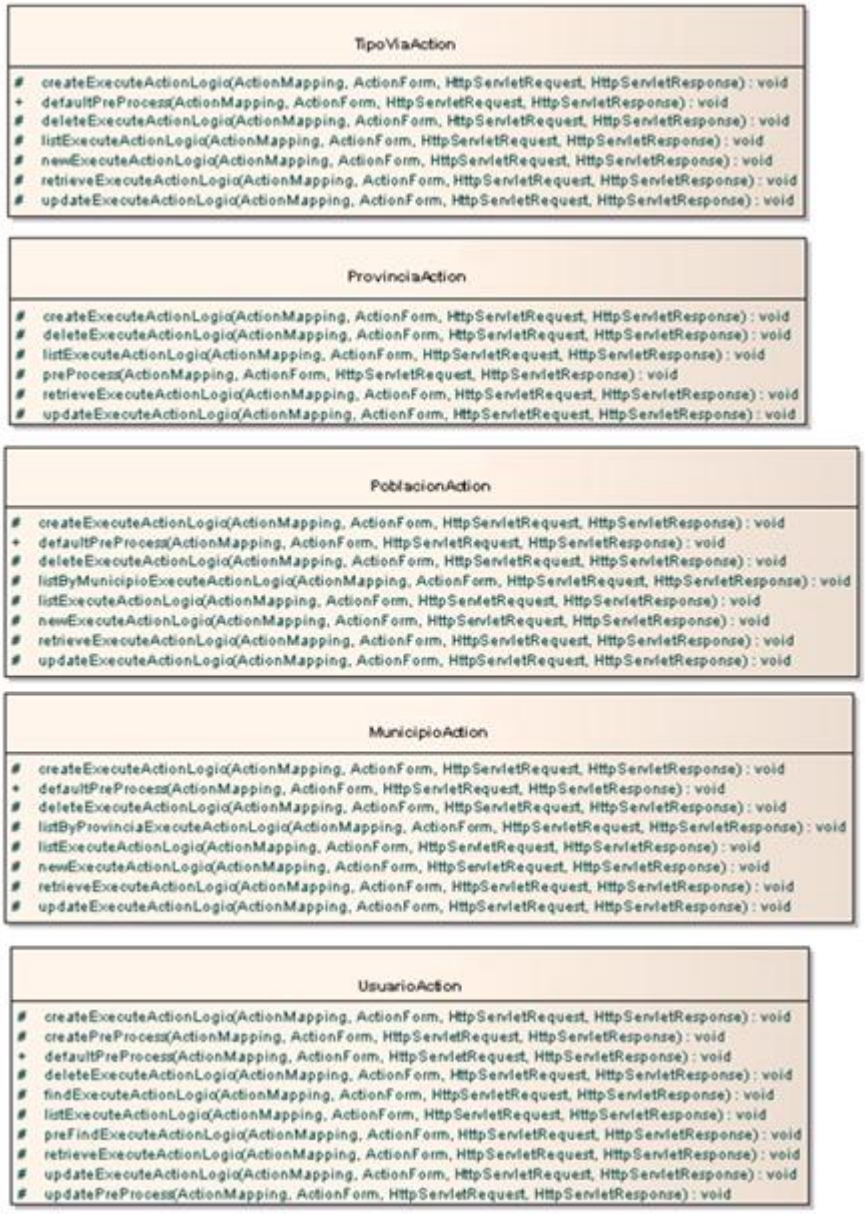


Ilustración 35 - Clases Action del subsistema Administración

5.2.1.5.2 Interfaces Manager

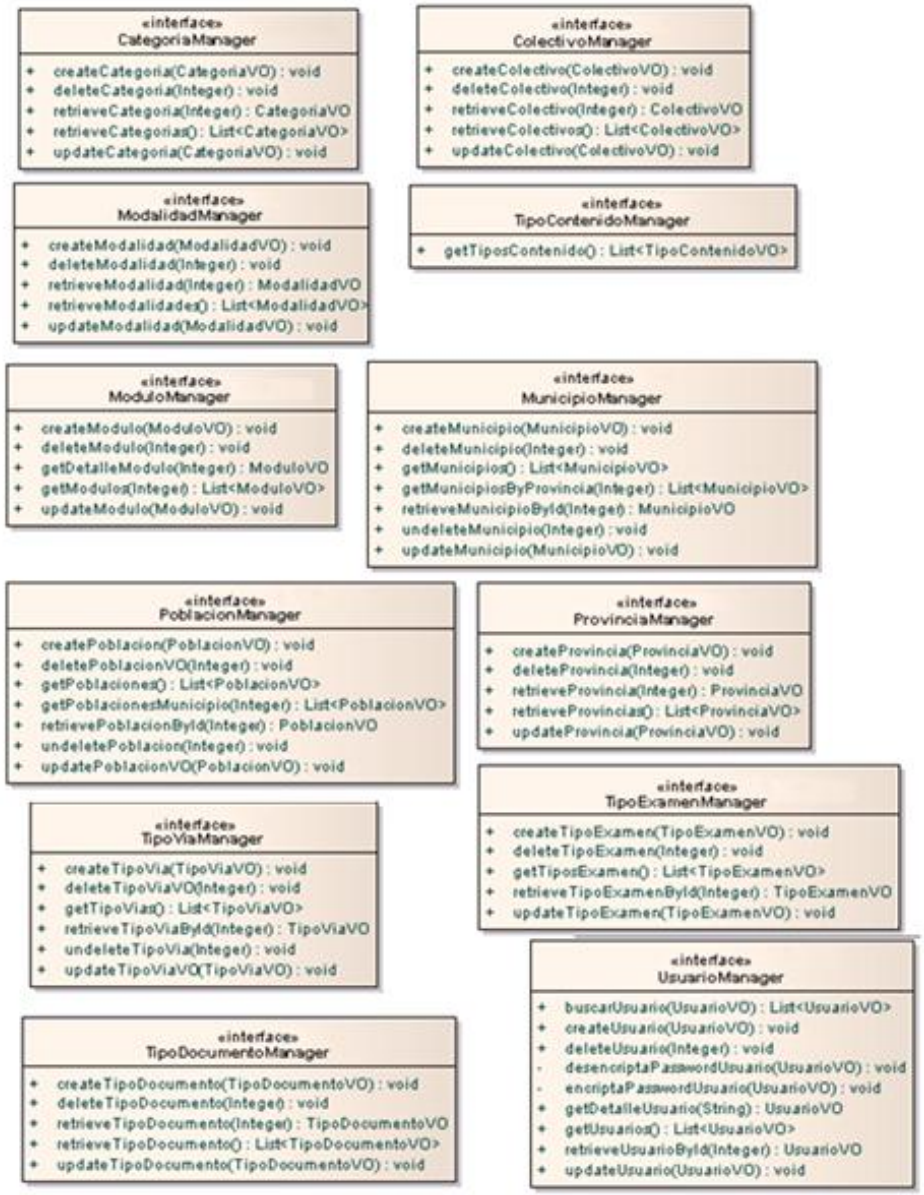


Ilustración 36 - Interfaces Manager del subsistema Administración

5.2.1.5.3 Interface DAO

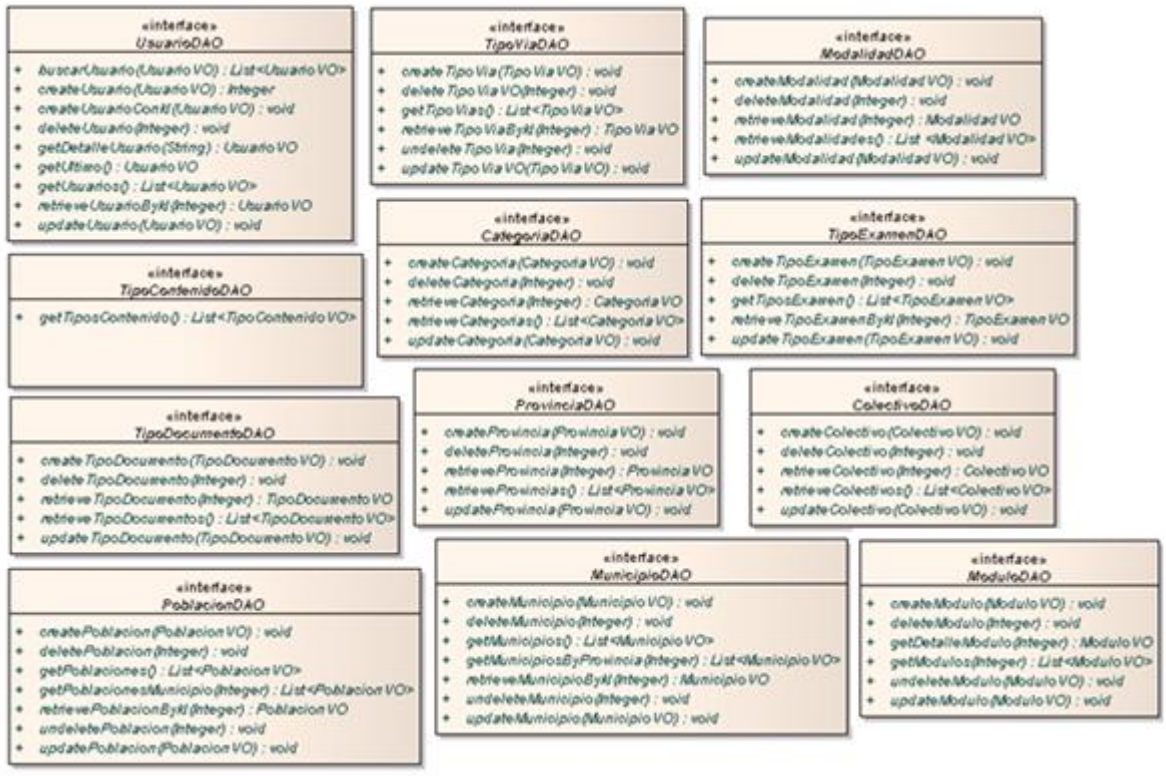


Ilustración 37 - Interfaces DAO del subsistema Administración

5.3 Diseño físico de datos

Como sistema gestor de base de datos se ha seleccionado MySQL, el cual proporciona un servidor de base de datos SQL (Structured Query Language) muy rápido, multi-threaded, multi usuario y robusto. El servidor MySQL está diseñado para entornos de producción críticos, con alta carga de trabajo.

La versión del servidor de la base de datos será MySql Server.

Las siguientes herramientas nos ayudarán con el tratamiento de la base de datos:

- **MySql Administrator**: Gestión de usuarios y permisos. Copia y respaldo de la base de datos.
- **MySql Query Browser**. Entorno para lanzar sentencias SQL de creación, manipulación y consulta sobre la base de datos.

A continuación podemos ver el diseño final de la base de datos. La codificación utilizada para la base de datos es UTF8.

Texto
OpenDocument

1 Script SQL

6. Plan de pruebas

6.1 Pruebas de caja blanca

En este documento se especifican pruebas de caja blanca sobre algunas de las funciones de la aplicación. En concreto, se han seleccionado aquellas más significativas o que se han considerado más importantes para el funcionamiento de la aplicación, puesto que ninguna de ellas tiene una complejidad elevada. También se indica, durante el desarrollo de las pruebas, que hay funciones que tienen una gran similitud, en estos casos se detalla el conjunto de pruebas de caja blanca para una de ellas entendiendo que para las demás es similar.

Para cada una de las funciones se realizan distintos subapartados: en el primero de ellos se incluye el código de la función, en el segundo el grafo asociado a dicha función, en el tercero la complejidad ciclomática del grafo y, finalmente, en el cuarto apartado, se incluye el conjunto básico de caminos independientes y los casos de prueba que validan esos caminos, es decir, el conjunto de valores concretos que permiten la ejecución de cada camino básico.

6.1.1 Método "updateExecuteActionLogic" en "EdicionAction"

Esta función pertenece a la unidad "Ediciones".

6.1.1.1 Código de la función

```
protectedvoid updateExecuteActionLogic(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
1logger.debug("****** Ejecutando updateExecuteActionLogic *******");
1EdicionForm edicionForm = (EdicionForm) form;
1 EdicionVO edicionVO = new EdicionVO();
1 edicionForm.populate(edicionVO);
1 EdicionVO edicionConCurso = (EdicionVO) edicionVO;
1 edicionDelegate.updateEdicion(edicionConCurso);
1 HorarioVO horario =

      horarioDelegate.retrieveHorarioEdicion(edicionVO.getIdEdicion())
      ;
 if(2horario != null) {
3 horario.setFechaInicio(edicionVO.getFechaInicio());
3 horario.setFechaFin(edicionVO.getFechaFinalizacion());
```

```
3 horarioDelegate.updateHorario(horario);
4}
5 //Para mostrar el mensaje de éxito
5 request.setAttribute(RequestKeys.EXITO_OPERACION, 0);
 5((EdicionForm) form).set(new EdicionVO());
}6
```

6.1.1.2 Grafo asociado a la función

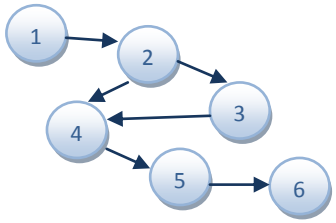


Ilustración 38–Grafo: "update" en "EdicionAction"

6.1.1.3 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene un predicado por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de dos**.

Si lo que se observa es el grafo, en él se pueden contar hasta dos regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de seis aristas que unen los seis nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 6 – 6 + 2 es igualmente dos, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.1.4 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1. (1, 2, 4, 5, 6)
2. (1, 2, 3, 4, 5, 6)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida

camino	Casos de prueba	Salida esperada	Salida obtenida
1	Aún no hay horario para la edición.	Se actualiza la edición, y se muestra el mensaje de éxito.	La esperada.
2	Ya hay un horario para la edición.	Se actualiza la edición y las fechas de inicio y fin en el horario de la misma. Se muestra el mensaje de éxito.	La esperada.

6.1.2 Método "deleteExecuteActionLogic" en "ActualizarInstructores"

Esta función pertenece a la unidad "Ediciones".

6.1.2.1 Código de la función

```
protectedvoid actualizarExecuteActionLogic(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
1logger.debug("****** Ejecutando actualizarExecuteActionLogic *****");
1 Integer idEdicion = ((EdicionVO) request.getSession()

.getAttribute(SessionKeys.EDICION)).getIdEdicion();
  //Recuperamoslalistade instructoresEdicion resultadodelabúsqueda:
1List<InstructorEdicionVO> listaInstructores =
 (List<InstructorEdicionVO>)request.getSession()
.getAttribute(SessionKeys.LISTA_INSTRUCTORES_BUSQUEDA);
1int tam = listaInstructores.size();

for(int i=0; 2 i<tam; i++) {
3 String idInstructor =
                listaInstructores.get(i).getIdInstructor().toString()
                ;
3 String precioDieta = request.getParameter("precioDieta"
                                                        +
idInstructor);
3 String precioKilometraje = request.getParameter("precioKilometraje"
                                                        +
idInstructor);
3 String cocheEmpresa = request.getParameter("cocheEmpresa"
                                                        +
idInstructor);
3 String tipoInstructor = request.getParameter("tipoInstructor"
                                                        +
idInstructor);
3 InstructorEdicionVO ieVO = new InstructorEdicionVO();
3 ieVO.setIdInstructor(new Integer(idInstructor));
3 ieVO.setPrecioDieta(new Double(precioDieta));
3 ieVO.setPrecioKilometraje(new Double(precioKilometraje));
3 ieVO.setCocheEmpresa(new Integer(cocheEmpresa));
3 ieVO.setTipoInstructor(new Integer(tipoInstructor));
3 ieVO.setIdEdicion(idEdicion);
```

```
     //Ya tenemos todos los datos en el VO, ahora lo actualizamos:
3 instructorEdicionDelegate.updateAsignarInstructor(ieVO);
 }4 //fíndel for.

 //Paramostrar el mensaje de éxito:
5 request.setAttribute(RequestKeys.EXITO_OPERACION, 0);
//Actualizar el listado resultante de la búsqueda:
5 buscarActualizarExecuteActionLogic(mapping,form,request,response);
}6
```

6.1.2.2 Grafo asociado a la función

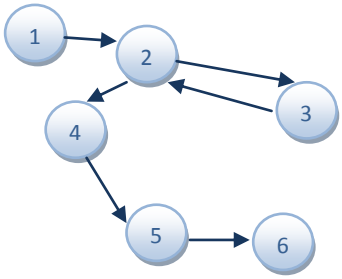


Ilustración 39–Grafo: "actualizar" en "ActualizarInstructoresAction"

6.1.2.3 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene un predicado por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de dos**.

Si lo que se observa es el grafo, en él se pueden contar hasta dos regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de seis aristas que unen los seis nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 6 – 6 + 2 es igualmente dos, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.2.3.1 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1. (1, 2, 3, 2, 4, 5, 6)
2. (1, 2, 4, 5, 6)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida
1	Se modifican los precios de dieta, de kilometraje, si precisa coche de empresa o no, y/o el rol del instructor para una edición concreta.	Los datos de los instructores/as modificados para la edición, son actualizados correctamente. Se muestra mensaje de éxito.	La esperada.
2	No hay instructores asociados a la edición. Por lo tanto, no hay datos que actualizar.	No aparece el botón para actualizar datos de los instructores, y el listado se encuentra vacío.	La esperada.

6.1.3 Método "deleteExecuteActionLogic" en "CursoAction"

Esta función pertenece a la unidad "Cursos".

6.1.3.1 Código de la función

```
protectedvoid deleteExecuteActionLogic(ActionMapping mapping,
      ActionForm form, HttpServletRequest request,
                  HttpServletResponse response) throws Exception
{
1logger.debug("***********      Ejecutando      deleteExecuteActionLogic
**********");
1Integer idCurso = new

      Integer((String)request.getParameter(RequestKeys.ID_CURSO));
//Ver si se puede borrar:
1List<String> error = new ArrayList<String>();
1 List<EdicionVO> listaEdicionesCurso =
                                    edicionDelegate.edicionesCurso(idCu
                                    rso);
if(2listaEdicionesCurso!=null&&3 !listaEdicionesCurso.isEmpty()) {
  //Ya se han creado ediciones para ese curso: NO BORRAR
 4 error.add("error.borrarCurso.conEdiciones");
 4 saveErrors(error);
}else { //Sí se puede borrar:
  //Quiero saber si el curso que voy a borrar es homologado o no, para
borrar tb su homologación si procede:
 5 CursoVO cursoAux = cursoDelegate.retrieveCurso(idCurso);
 5 cursoDelegate.deleteCurso(idCurso);
 5 Integer tipoCurso = cursoAux.getTipoCurso();
  //Si el tipo de curso es 0: no es homologado. Si es 1 sí lo es:
      if (6 tipoCurso.equals(1)) {
            7 cursoHomologadoDelegate.deleteCursoHomologado(idCurso);
      }8
}9
//Actualizar la vista:
//Recupero el VO de la búsqueda:
10 CursoVO cursoVO = (CursoVO)
10 request.getSession().getAttribute(SessionKeys.CURSO);
```

```
10 actualizaVista(cursoVO, request, form);
}11
```

6.1.3.2 Grafo asociado a la función

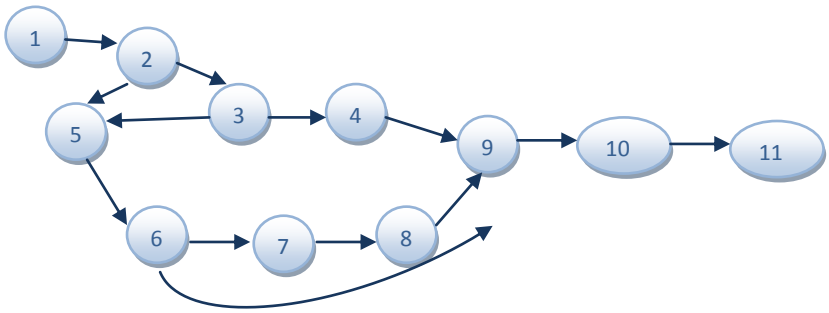


Ilustración 40–Grafo: "delete" en "CursoAction"

6.1.3.3 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene tres predicados por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de cuatro**.

Si lo que se observa es el grafo, en él se pueden contar hasta cuatro regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de trece aristas que unen los once nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 13 – 11 + 2 es igualmente cuatro, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.3.4 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1. (1, 2, 3, 4, 9, 10, 11)
2. (1, 2, 5, 6, 8, 9, 10, 11)
3. (1, 2, 5, 6, 7, 8, 9, 10, 11)
4. (1, 2, 3, 5, 6, 8, 9, 10, 11)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida
1	Ya hay ediciones del curso.	No se puede borrar el curso, y se muestra el mensaje de error.	La esperada.
2	Aún no hay ediciones del curso, y el curso no es homologado.	Se borra el curso, y se muestra el mensaje de éxito.	La esperada.
3	Aún no hay ediciones del curso, y el curso es homologado	Se borra el curso y los datos de su homologación. Se muestra el mensaje de éxito.	Se borra el curso. Se muestra el mensaje de éxito.
4	El listado de ediciones del curso es vacío, y el curso es homologado	Se borra el curso y los datos de su homologación. Se muestra el mensaje de éxito.	Se borra el curso. Se muestra el mensaje de éxito.

6.1.4 Método "matricularActionLogic" en "MatricularNoPreinscritosAction"

Esta función pertenece a la unidad "Ediciones". Totalmente análoga a matricular alumnos preinscritos.

6.1.4.1 Código de la función

```
protectedvoid matricularExecuteActionLogic(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
1logger.debug("** Ejecutando matricularNOPreinscritosExecuteAction ");
1 String matriculaciones[]=null;
  //Recojemos los alumnos marcados de la página actual
1 matriculaciones = request.getParameterValues("idAlumno");
1Integer idEdicion = (Integer)request.getSession()
.getAttribute(SessionKeys.ID_EDICION);
for(int i=0;2 i<matriculaciones.length; i++){
3 EdicionAlumnosEstadoVO eamVO = new EdicionAlumnosEstadoVO();
3eamVO.setIdAlumno(new Integer(matriculaciones[i]));
3eamVO.setIdEdicion(idEdicion);
3eamVO.setIdEstado(new Integer ("1"));
3 edicionAlumnosEstadoDelegate.createEdicionAlumnosEstado(eamVO);
3EdicionVO edicionVO = (EdicionVO)
                  request.getSession().getAttribute(SessionKeys.EDICION
            )
3Integer idCurso = edicionVO.getCursoVO().getIdCurso();
3 List<RequisitoVO> listaReqCurso =

                  cursoDelegate.retrieveListaReqCurso(idCurso);
   if (4 !listaReqCurso.equals(null) ) {
      5  GregorianCalendar calendario = new GregorianCalendar();
      5  Integer año = calendario.get(Calendar.YEAR);
      5  Integer mes = calendario.get(Calendar.MONTH)+1;
      5  Integer dia = calendario.get(Calendar.DAY_OF_MONTH);
      5  String fecha = dia+"/"+mes+"/"+año;
```

```
      5 SimpleDateFormat sdf= new SimpleDateFormat("dd/MM/yyyy");
  for (int index=0; 6 index<listaReqCurso.size(); index++) {
      7 ReqAlumno raVO = new ReqAlumno();
      7raVO.setIdRequisito(listaReqCurso.get(index).getId());
7 raVO.setIdAlumno(new Integer(matriculaciones[i]));
      7 raVO.setIdEdicion(idEdicion);
      7 raVO.setCumple(0);
      7 raVO.setFecha(sdf.parse(fecha));
      7 reqAlumnoDelegate.createReqAlumno(raVO);
     }8
    }9//Finif
  }10//Finfor
}11
```

6.1.4.2 Grafo asociado a la función

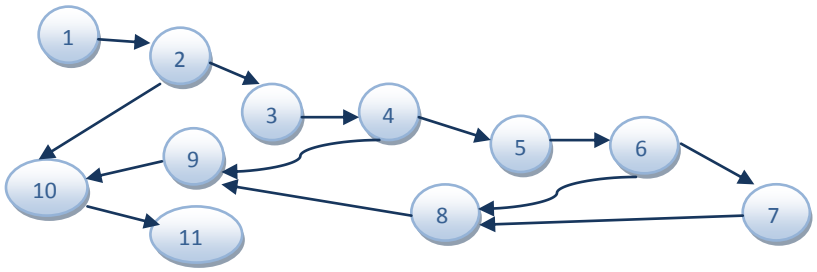


Ilustración 41–Grafo: "matricular" en "MatricularNoPreinscritosAction"

6.1.4.3 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene tres predicados por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de cuatro**.

Si lo que se observa es el grafo, en él se pueden contar hasta cuatro regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de trece aristas que unen los once nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 13 – 11 + 2 es igualmente cuatro, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.4.4 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1. (1, 2, 10, 11)
2. (1, 2, 3, 4, 9, 10, 11)
3. (1, 2, 3, 4, 5, 6, 8, 9, 10, 11)
4. (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida
1	No se ha seleccionado ningún alumno/a para matricular, pero se pulsa el botón matricular.	Mostrar mensaje que indique que se ha de seleccionar algún alumno/a.	La página se queda en blanco.
2	Se matriculan alumnos en una edición que no tiene requisitos.	El alumno/a queda matriculado, y se muestra en el listado de alumnos/as matriculados.	La esperada.
3	Se matriculan alumnos en una edición, cuyo listado de requisitos existe, pero está vacío.	El alumno/a queda matriculado, y se muestra en el listado de alumnos/as matriculados.	La esperada.
4	Se matriculan alumnos en una edición, con requisitos.	El alumno/a queda matriculado, y se muestra en el listado de alumnos/as matriculados. Se crean los requisitos para ese alumno.	La esperada.

6.1.5 Método "inscribirExecuteActionLogic" en "PreinscribirAction"

Esta función pertenece a la unidad "Ediciones".

6.1.5.1 Código de la función

```
protectedvoid inscribirExecuteActionLogic(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception {
1logger.debug("**** Ejecutando inscribirExecuteActionLogic *****");
1String solicitudes[]=null;
//Recojemos los alumnos marcados de la página actual
1 solicitudes = request.getParameterValues("idAlumno");
for(int i=0;2 i<solicitudes.length; i++){
      3Integer idEdicion = (Integer)
            request.getSession().getAttribute(SessionKeys.ID_EDICION);
      3 EdicionAlumnosEstadoVO eamVO = new EdicionAlumnosEstadoVO();
      3 eamVO.setIdAlumno(new Integer(solicitudes[i]));
      3 eamVO.setIdEdicion(idEdicion);
      3 eamVO.setIdEstado(new Integer ("2"));
3edicionAlumnosEstadoDelegate.createEdicionAlumnosEstado(eamVO);
  } 4
5 request.getSession().setAttribute(
```

```
SessionKeys.LISTA_ALUMNOS_NO_PREINSCRITOS,
new LinkedList<AlumnoVO>());
5 Integer idEdicion = (Integer)request.getSession()

            .getAttribute(SessionKeys.ID_EDICION);
5 request.getSession().setAttribute(
SessionKeys.LISTA_ALUMNOS_PREINSCRITOS,
                alumnoDelegate.listaAlumnosPreinscritos(idEdicion));
5 preInscribirExecuteActionLogic(mapping, form, request, response);
}6
```

6.1.5.2 Grafo asociado a la función

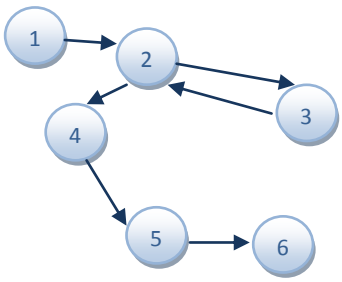


Ilustración 42–Grafo: "inscribir" en "PreinscribirAction"

6.1.5.3 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene un predicado por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de dos**.

Si lo que se observa es el grafo, en él se pueden contar hasta dos regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de seis aristas que unen los seis nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 6 – 6 + 2 es igualmente dos, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.5.4 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1. (1, 2, 3, 2, 4, 5, 6)
2. (1, 2, 4, 5, 6)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida
1	Se marca uno o varios alumnos/as para preinscribir en una edición.	Los alumnos/as quedan preinscritos/as en la edición, y se actualiza el listado de alumnos/as preinscritos.	La esperada.
2	No se marca ningún alumno/a para preinscribir, pero se pulsa el botón.	Vuelve a mostrar la misma página, manteniendo el filtrado de alumnos/as.	La esperada.

6.1.6 Método "cambiarEstado" en "MatriculadosCambiarEstadoAction"

Esta función pertenece a la unidad "Ediciones".

6.1.6.1 Código de la función

```
protectedvoid cambiarEstadoExecuteActionLogic(ActionMapping
mapping,ActionForm form, HttpServletRequest request,
HttpServletResponse response) throws Exception
{
1 logger.debug("*** Ejecutando cambiarEstadoExecuteActionLogic ****");
1 Integer idEdicion = (EdicionVO)request.getSession()
                    .getAttribute(SessionKeys.EDICION)).getIdEdicion();
1 Map<String,Integer>map = new HashMap<String,Integer>();
1map.put("idEdicion", idEdicion);
  //Recuperamos la lista de alumnos matriculados o de baja:
1List<EdicionAlumnosEstadoVO> listaAlumnos =
                      (List<EdicionAlumnosEstadoVO>)
request.getSession()

      .getAttribute(SessionKeys.LISTA_ALUMNOS_ESTADO);
1 int tam= listaAlumnos.size();
for(int i=0; 2 i<tam;i++){
      3 String idAlumno = listaAlumnos.get(i).getIdAlumno().toString();
      3 String idEstado= request.getParameter("idEstado" + idAlumno);
      //Rellenamos el Map conlosdatos a actualizar:
      3 Map<String, Integer> mapActualizar =
new HashMap<String, Integer>();
      3                                  mapActualizar.put("idAlumno",
listaAlumnos.get(i).getIdAlumno());
      3 mapActualizar.put("idEdicion", idEdicion);
      3 mapActualizar.put("idEstado", new Integer(idEstado));
      //Ya tenemos todos los datos en el VO, ahora lo actualizamos:
      3 eaeDelegate.updateEdicionAlumnosEstado(mapActualizar);
  } 4 //fíndel for
   //Recuperamos la listade alumnos matriculados o debaja con el
   //estado ya actualizado:
5 Integer idEdicion = (Integer)request.getSession()

            .getAttribute(SessionKeys.ID_EDICION);
5 request.getSession().setAttribute(SessionKeys.LISTA_ALUMNOS_ESTADO,
```

```
eaeDelegate.listarYaMatriculados(idEdicion));
//Para mostrar el mensaje de éxito:
5 request.setAttribute(RequestKeys.EXITO_OPERACION, 0);
}6
```

6.1.6.2 Grafo asociado a la función

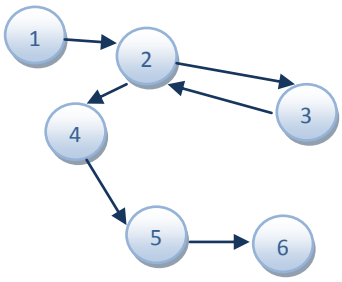


Ilustración 43–Grafo: "cambiarEstado" en "MatriculadosCambiarEstadoAction"

6.1.6.3 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene un predicado por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de dos**.

Si lo que se observa es el grafo, en él se pueden contar hasta dos regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de seis aristas que unen los seis nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 6 – 6 + 2 es igualmente dos, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.6.4 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1. (1, 2, 3, 2, 4, 5, 6)
2. (1, 2, 4, 5, 6)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida
1	Se cambia el estado de uno o varios alumnos/as matriculados en la edición.	El estado de los alumnos/as para esa edición, queda actualizado.	La esperada.
2	No hay ningún alumno/a matriculado/a en la edición.	Se muestra el listado vacío, y se indica que no hay alumnos/as matriculados. No se permite pulsar el botón de actualizar.	La esperada.

6.1.6.5 Método "asignarExecuteActionLogic" en "AsociarInstructoresAction"

Esta función pertenece a la unidad "Ediciones". Asigna instructores a una edición.

6.1.6.6 Código de la función

```
protectedvoid asignarExecuteActionLogic(ActionMapping mapping,
ActionForm form, HttpServletRequest request,
 HttpServletResponse response) throws Exception
{
1logger.debug("***** Ejecutando asignarExecuteActionLogic *******");
1 String asignaciones[]=null;
//Recojemos los instructores marcados de la página actual
1 asignaciones = request.getParameterValues("idInstructor");
1Integer idEdicion = (Integer) request.getSession()
                  .getAttribute(SessionKeys.ID_EDICION);
2 if(asignaciones!=null){
for(int i=0;3i<asignaciones.length; i++){
4 InstructorEdicionVO ieVO = new InstructorEdicionVO();
      4 ieVO.setIdInstructor(newInteger(asignaciones[i]));
      4 ieVO.setIdEdicion(idEdicion);
      4 InstructorVO instructorVO = instructorDelegate
                  .getDetalleInstructor(new Integer(asignaciones[i]));
      4ieVO.setPrecioDieta(instructorVO.getPrecioDieta());
      4 ieVO.setPrecioKilometraje(instructorVO.getPrecioKilometraje());
      5 if(instructorVO.getCocheEmpresa()==true) {
      6ieVO.setCocheEmpresa(new Integer(1));
      }else{
      7 ieVO.setCocheEmpresa(new Integer(0));
      } 8
      9Boolean borrado = false;
      9 ieVO.setBorrado(borrado);
      9 instructorEdicionDelegate.createAsignarInstructor(ieVO);
    } 10 //fin del for.
  } 11
}12
```

6.1.6.6.1 Grafo asociado a la función

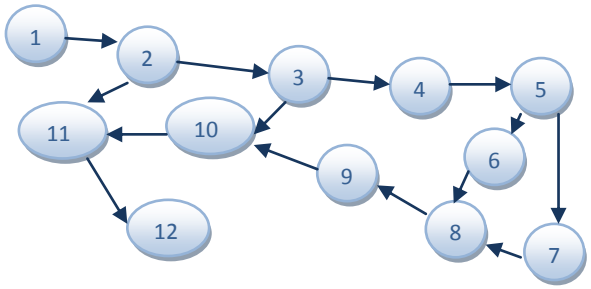


Ilustración 44–Grafo: "asignar" en "MatriculadosCambiarEstadoAction"

6.1.6.7 Complejidad ciclomática

Las líneas marcadas en color rojo y en formato negrita, son las condiciones del código de la función. Como se puede ver, la función tiene un predicado por lo que según la formula "Número de predicados + 1" la función tiene una **complejidad ciclomática de cuatro**.

Si lo que se observa es el grafo, en él se pueden contar hasta cuatro regiones distintas, lo cual también es indicativo de la complejidad ciclomática de la función.

Teniendo en cuenta el último de los métodos, el grafo tiene un total de catorce aristas que unen los doce nodos que se pueden observar. Aplicando la pertinente fórmula se concluye que 14 – 11 + 2 es igualmente dos, por lo que la complejidad ciclomática es la misma en todos los casos.

6.1.6.8 Caminos del grafo y conjunto de pruebas

Los caminos independientes del grafo son los siguientes:

1 (1, 2,11, 12)
2 (1, 2, 3, 10, 11, 12)
3 (1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12)
4 (1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12)

A continuación se muestra la tabla con los distintos casos de prueba para recorrer los caminos del grafo.

camino	Casos de prueba	Salida esperada	Salida obtenida
1	No se ha asignado ningún instructor a la edición, pero se pulsa el botón "Asignar".	No se asigna ningún instructor/a a la edición. Se vuelve a la misma página, en el estado anterior.	La esperada.
2	El listado de instructores/as a asignar a la edición existe, pero está vacío.	No se asigna ningún instructor/a a la edición. Se vuelve a la misma página, en el estado anterior.	La esperada.

camino	Casos de prueba	Salida esperada	Salida obtenida
3	Se asigna uno o varios instructores/as a la edición, que poseen coche de empresa.	Se asigna el instructor/a a la edición, y se marca que posee coche de empresa.	La esperada.
4	Se asigna uno o varios instructores/as a la edición, que no poseen coche de empresa.	Se asigna el instructor/a a la edición, y se marca que no posee coche de empresa.	La esperada.